D D WEBSTER ELECTRONICS PTY LTD
17 MALVERN STREET, BAYSWATER 3153, AUSTRALIA
Telephone: 729 8444 Telex: 36251

SPECTRUM ELEVEN

SERIES 'SS'

R T-11 SOFTWARE MANUAL

VERSION 2

This manual was supplied in conjunction with

SPECTRUM ELEVEN computer serial no.

Dated: September, 1982

WEBSTER ELECTRONICS

This manual describes only the software components applicable to the 'SS' Series Spectrum Eleven computers. Software information on the 'SP' series Spectrums can be found in the original Spectrum Eleven Users Manual, issued December, 1978.

Also available is a Spectrum Eleven Hardware Manual, issued in conjunction with this software manual.

SECTION 1 AVAILABLE SOFTWARE

1 SCOPE

1.1 MANUAL LAYOUT AND USE
1.2 SOFTWARE SUPPLIED

2 OPERATING SYSTEMS

2.1 RT-11 V4 Operating System
2.2 TSX
2.3 TSX-Plus
2.4 RSX-11M-PLUS
2.5 RSX-11M
2.6 RSTS/E
2.7 UNIX

3 RT-11 SUPPORT SYSTEMS

3.1 TSX-Net
3.2 RTSORT
3.3 VIRTUAL UNIT (VU) SYSTEM

4 RT-11 LANGUAGES

4.1 SINGLE-USER BASIC-11 (V.02) Interpreter
4.2 MULTI-USER BASIC-11 (V.02) Interpreter
4.3 RJ-11 COBOL
4.4 COBOL-Plus
4.5 FORTRAN IV (V2.0) Compiler
4.6 PASCAL-2 Compiler
4.7 C
4.8 MACRO-11

5 RT-11 GENERAL APPLICATIONS SOFTWARE

5.1 WP-SATURN
5.2 TEXT-11
5.3 SIMILE
5.4 VDT
5.5 FASTEXT

6 RT-11 SPECIAL APPLICATIONS SOFTWARE

6.1 MONECS
6.2 T-MONECS
6.3 SCHOLAR
6.4 T.P.C.M.
6.5 T.A.S.M.
6.6 PRACTICE MANAGEMENT SYSTEM

6.7 MANAGEMENT COMPUTING SYSTEMS
6.8 COMMERCIAL BUSINESS SYSTEMS
6.9 PIG MANAGEMENT SYSTEM
6.10 COMPUTER AIDED PLASTIC MOLDING SYSTEM
6.11 MEDIPAC
6.12 C.A.R.E.
6.13 SIMPL
6.14 LEGAS
6.15 PROPERTY MANAGEMENT SYSTEM
6.16 SATURN-CALC

SECTION 2 RT-11 INSTALLATION GUIDE

1 RT-11 OPERATING GUIDE

1.1 SYSTEM CONFIGURATION
1.2 STARTING THE RT-11 COMPUTER SYSTEM
1.3 INTERACTING WITH THE RT-11 COMPUTER SYSTEM
1.4 STORAGE VOLUMES
1.5 SYSTEM SECURITY
1.6 RUNNING AN APPLICATION PROGRAM
1.7 ABORTING AN APPLICATION PROGRAM

2 SYSTEM DISKS

2.1 RT-11 LICENCES
2.2 RT-11 SYSTEM FILE DESCRIPTIONS

2.2.1 Indirect Startup Command File
2.2.2 RT-11 System Files & Device Handlers
2.2.3 Memory Image Files
2.2.4 Executable Foreground Program File
2.2.5 System Macro library
2.2.6 Object Output File/Library File
2.2.7 TECO Files

2.3 SYSTEM DISK DESCRIPTION
2.4 SYSTEM DISK CREATION
2.5 FILE MAINTENANCE

2.5.1 Floppy Disks
2.5.2 Cartridge Disks
2.5.3 Winchester Disks

2.6 RT-11 COMMAND CONSIDERATIONS

2.6.1 Copying Procedures
2.6.2 Volume Size Squeeze

3 **RT-11 SYSTEM CONVENTIONS**

3.1 FILES
3.2 FILE PROTECTION PROCEDURES
3.3 INDIRECT FILES

3.3.1 Creating an Indirect File
3.3.2 Executing an Indirect File

3.4 DEVICE HANDLERS
3.5 DEVICE STRUCTURES
3.6 PHYSICAL DEVICE NAMES
3.7 RT-11 SYSTEM MESSAGES

SECTION 3 RUN-TIME RT-11 USER GUIDE

1 **INTRODUCTION TO RT-11**

1.1 MONITOR
1.2 UTILITIES AND PROGRAMS
1.3 LANGUAGE PROCESSORS

2 **RT-11 LANGUAGE PROCESSORS**

2.1 INTRODUCTION

3 **RT-11 MONITORS**

3.1 RT-11 MONITOR COMPONENTS
3.2 KMON COMMAND FORMAT
3.3 CONVENTIONS AND RULES
3.4 WILDCARDS
3.5 KEYBOARD MONITOR COMMANDS

3.5.1 Index of Commands
3.5.2 KMON Commands

SECTION 4 STANDARD RUN-TIME RT-11 UTILITIES

1 **COMMAND STRING INTERPRETER**

1.1 SYNTAX

2 **RT-11 UTILITIES & PROGRAMS**

2.1 SYSTEM UTILITIES

2.1.1 Peripheral Interchange Program
2.1.2 Device Utility Program
2.1.3 The Directory Program
2.1.4 File Exchange Program
2.1.5 Resource Utility Program

3 **TECO**

3.1 TECO COMMANDS
3.2 TECO COMMAND STRUCTURE
3.3 TECO COMMAND SUMMARY
3.4 IMMEDIATE ACTION CONTROL COMMANDS
3.5 USE OF TECO WITH RT-11

3.5.1 Startup
3.5.2 File Specification
3.5.3 Backup Files
3.5.4 Re-enter and Close
3.5.5 File Recovery
3.5.6 System Crash Recovery

3.6 COMMON ERROR MESSAGES
3.7 SAMPLE EDITING JOB

4 **EDIT**

4.1 EDIT KEY COMMANDS
4.2 EDIT CHARACTERISTICS
4.3 EDIT COMMAND STRUCTURE
4.4 EDIT COMMAND SUMMARY
4.5 GENERAL USE OF EDIT
4.6 COMMON ERROR MESSAGES

SECTION 5 SPECTRUM RT-11 OPERATOR UTILITIES

1 **FUTIL**

2 **BACKUP**

3 **BAKFLP**

4 **SAVRES**

SECTION 6 SPECTRUM RT-11 ENGINEERING UTILITIES

1 **WUTIL**

2 **RANDY**

3 **MAGTAPE BOOTSTRAP PROCEDURES**

SECTION 7 SPECTRUM RT-11 DEVICE HANDLERS

1 **DEVICE HANDLERS**

1.1 CARD READER HANDLER
(Not documented - under revision)

1.2 ITERATOR HANDLER
1.3 FLOPPY DISK HANDLER
1.4 WINCHESTER DISK HANDLER
1.5 SPECTRUM MAGTAPE HANDLER

INTRODUCTION

Software available for Spectrum Eleven computers covers a diverse range of applications. These include DIGITAL operating systems and their pertinent components such as text editors and utilities, various programming languages, general and special applications software, and utilities written specifically for Spectrum machines.

This manual describes the Spectrum specific features of the RT-11, TSX and TSX-Plus operating systems, and includes details of all related software. Documentation for other operating systems is available from the originator of the system(s).

As software is always subject to evaluation and improvement, modifications made after the date of issue of this manual, are not reflected in this version.

1.1 MANUAL LAYOUT AND USE

This manual is structured into seven major divisions; each being sub-divided into chapters from 1 thru 'n'.

Section 1 describes all available software, and where necessary highlights reference material available.

Section 2 instructs the user how to use RT-11 in conjunction with the Spectrum Eleven computer.

Section 3 is a complete reference for run-time users of RT-11.

Section 4 describes the run-time RT-11 utilities available.

Section 5 includes documentation on all Spectrum support utilities. These utilities provide for save/restore routines, formatting, copying and testing.

Section 6 includes documentation on all Spectrum engineering utilities. These utilities must be used with extreme caution if available on the distribution disk.

Section 7 describes the non-standard DEC RT-11 device handlers used on Spectrum Eleven computers.

Where reference is made to another Section of this manual, the Section and the Chapter number are detailed. Where reference is made between chapters in the same section, the Section will not be specified and where reference is made to the Spectrum Eleven Hardware Manual, the chapter/section number is not specified.

1.2 SOFTWARE SUPPLIED

All Spectrum computers are supplied complete with:

DEC RT-11 Version 4 Run-time operating system which excludes MACRO, LINK, ODT and handbooks

Webster support utilities, which are dependent on hardware configurations, may include:

FUTIL Spectrum floppy disk formatting and copying utility

BACKUP Spectrum floppy disk copying utility providing disk identification

BAKFLP Spectrum Winchester to floppy disk save/restore utility

SAVRES Spectrum Winchester to magtape/floppy disk, save/restore utility

Webster engineering function utilities which may include:

WUTIL Winchester disk verification and maintenance utility

RANDY Universal disk exerciser for checking random access device hardware

Magtape bootstrap procedures

INTRODUCTION

Spectrum Eleven computers support the running of several operating systems; the selected system being dependent on user's requirements along with the proposed hardware to be installed.

These Spectrum compatible operating systems are briefly described in this chapter and where applicable, references are given for those systems not covered by this manual.

2.1 RT-11 V4 Operating System

Full licence system which includes MACRO, handbooks, etc.

RT-11 is a single-user disk operating system designed for interactive program development and on-line applications. It supports 256 Kb of main memory, three compatible monitors, and a variety of program development and system utilities. Optional support includes a number of high-level language processors.

RT-11 is designed for the single, interactive user but can provide multi-user capabilities when coupled with TSX or TSX-Plus. The English-language keyboard commands are easy to use and understand, and command sequences can be stored in an indirect command file to be invoked repeatedly by the user.

Other features include: contiguous file structure for random-access devices, device-independent I/O programming, three modes of I/O operation (synchronous, asynchronous, event-driven), and the capacity for RT-11 to interface very easily with unique drivers merely by writing a device handler and storing it as a file on the system device.

A run-time RT-11 licence (RT2) is available for processing application programs along with file maintenance functions. With the exception of compilers and interpreters which do not need MACRO and LINK, program development facilities are not included.

Handbooks are not supplied with the run-time licence but the RT-11 system has been adequately documented (refer Section 3) for users of the run-time system.

2.2 TSX

TSX adds multi-user capabilities to a single-user RT-11 operating system and allows time-sharing facilities for up to 10 users. Minimum requirements include at least 56Kb of memory, a program swapping disk, and a line frequency clock. It is recommended that systems installed with more than 64Kb addressable memory use TSX-Plus. Memory management facilities are not required for TSX.

The TSX system operates under a standard unmodified RT-11 foreground/background or single-job monitor. When used with the FB monitor, a real-time job can be run in the foreground while TSX is run in the background. The background program space is swapped to and from a disk file on a service priority and time-slice basis to allow many users to share this memory area.

User application programs written for RT-11 can, in most cases, be run under TSX without change. However, a modified version of ODT must be used to debug programs under TSX, the RT-11 BATCH facility is not supported, and the PATCH program cannot be run.

Features of TSX that are common with TSX-Plus are documented in the next paragraph.

2.3 TSX-Plus

TSX-Plus is a major enhancement of the TSX operating system and can take full advantage of memory up to 256Kb at the current version 2.2. In contrast to TSX, which runs in conjuntion with RT-11 to provide a time-sharing system, TSX-Plus supports RT-11 system service calls as its basic mode of operation. This results in lower system overhead and substantially improved performance over systems that must emulate RT-11 services.

TSX-Plus allows up to 16 users to use the system in a time-sharing environment with the advantage that many users can be resident in core concurrently. A mandatory requirement is an SS23 processor with memory management and a minimum of 128Kb. Memory is partitioned, allowing a maximum address space of 56Kb for each of the approximately 20 jobs that the system will support. A priority and time-slice system is used to control the jobs to remain in memory and the jobs to be swapped to a disk file.

Most RT-11 programs can be used with TSX-Plus without change or even having to be relinked. TSX-Plus supports MACRO-11, FORTRAN-IV, BASIC-11, PASCAL, and COBOL-Plus, and supports RT-11 utility programs such as PIP, DIR, DUP, TECO, etc. The TSX-Plus keyboard commands are an extended set of those provided by RT-11 and include the CCL commands such as COMPILE, EXECUTE, COPY, DELETE, DIR, etc..

The main features of TSX-Plus (and TSX) can be ascribed to:

Virtual lines that allow one time-sharing user to control several simultaneously running programs from a single terminal.

A transparent spooling system that provides fully automatic spooling output to one or more output devices.

A shared file record locking facility that provides controlled access to a common file being accessed and updated by more than one user.

A program performance monitor system.

Command files that allow parameters.

Remote dial-up lines and hardware are supported for development and installation of changes or system modifications.

A message transmission facility that provides a communication network for messages to be transferred between programs.

Advanced terminal support provides convenient operation with a wide variety of terminals.

User access control which provides controlled access to a system.

2.4 RSX-11M-PLUS

RSX-11M-PLUS is a moderate sized real-time multiprogramming system, requiring a minimum memory of 256Kb, that is optimized for larger multipurpose real-time applications and program development systems. This operating system, which is the top-end of the RSX-11M family, handles batch job execution and offers management of up to 250 priority levels, interactive program development and execution, multiuser multiprogramming, and good program protection.

Features unique to RSX-11M-PLUS include:

User mode I/D space which means that a user task has the ability to address up to 64 kilobytes of instruction and 64 kilobytes of data at the same time, giving a total of 128 kilobytes. This simplifies the development and enhances performance of large application programs by reducing the need for overlays.

A multistream batch processing capability which is controlled by operator intervention.

I/O request queue optimization which allows request queues for disks to be sorted by cylinder number of the request. The

average seek time is reduced, improving throughput.

Various performance enhancements which include priority scheduling, the ability to lock tasks in memory, and communication microprocessors to reduce the system load of interprocessor communications.

Features common to RSX-11M-PLUS and RSX-11M include:

Multiprogramming which is accomplished by logically dividing memory into a number of named partitions and all partitions in the system can then operate in parallel. This allows a number of programs to run simultaneously and maximizes the use of the central processor.

Support for MACRO-11, BASIC, COBOL, FORTRAN-IV, and APL. These systems also have a powerful data base management system, and a file system (FILES-11) that combines with the Record Management Services (RMS) or the File Control Services (FCS) to alleviate much of the work generally associated with creating files.

Data and system integrity which are maintained by use of user identification codes, privileged and non-privileged accounts, password protection, and separate user directories.

A comprehensive selection of application tools designed to shorten the program development cycle. These include PIP, SLP, LBR, ZAP, PAT, and BRU.

An optional Forms Management system (FMS) where programmers can create applications that use forms displayed on the terminal screen to handle user inquiry and response.

System utilities that give a choice of two editors, a sophisticated task builder to link modules and prepare executable programs, and online debuggers that allow users to examine, alter, search and execute programs.

The RSX-11M-PLUS operating system is not covered by this manual.

2.5 RSX-11M

The RSX-11M operating system has a subset of the capabilities of RSX-11M-PLUS and is a small to moderate sized real-time multiprogramming system. It can support up to 56Kb (without memory management) and 248Kb (with memory management).

It is designed to support applications demanding immediate response, such as factory automation, graphics, communications, and process monitoring.

Features of RSX-11M common to RSX-11M-PLUS, are described in the preceding paragraph.

The RSX-11M operating system is not covered by this manual. Reference RSX-11M/RSX-11S Documentation Directory, Digital Equipment Corporation AA-2593E-TC, which is a guide to the documentation set available for this system.

2.6 RSTS/E

RSTS/E is a moderate to large timesharing system that can support up to 63 concurrent jobs, including interactive terminal user jobs, detached jobs, and batch processing, and it provides excellent response time to users, who can number up to 63 at one time.

RSTS/E requires a minimum memory of 128Kb. Users can expect efficient operation as the operating system dynamically allocates processor time, memory space, file space, and peripherals to best suit changing demands.

RSTS/E supports MACRO, BASIC-PLUS (included), FORTRAN-IV, COBOL, and APL. It also has excellent security facilities which allow for privileged/unprivileged jobs and accounts, and includes system accounting facilities so that billing for time-sharing time can be accurately calculated.

Other features include:

Record Management Services (RMS) software that supports building and access of sequential, relative, and multikey indexed sequential file structures, and relieves programmers of many tedious tasks of I/O management.

System management utilities which include: system initialization and maintenance programs, resource management and accounting programs, system error logging and analysis programs, operator services and spooling programs, and user communication programs.

General system utility programs which include system information and terminal utility programs, file utility programs, and special service programs.

SYS system function calls which allow a BASIC-PLUS user program to perform special I/O functions, to establish special characteristics for a job, to set terminal characteristics, and to request execution of special monitor operations.

The RSTS/E operating system is not covered by this manual. Reference RSTS/E V7.0 Documentation Directory, Digital Equipment Corporation AA-2642D-TC, which is a guide to the documentation set available for this system.

2.7 UNIX

UNIX is powerful although relatively small operating system. It is written in a high level programming language called C, and most of the source code and documentation is available on-line.

The important aspect of UNIX is that it is constructed out of a small, basic set of concepts and programming modules, with a flexible method for interconnecting existing modules to make new functions. All system objects, including all I/O channels, look like files. It is therefore possible to cause input and output for almost any program to be taken from or to go to files, terminals, or other devices, at any time, without any particular planning on the part of the module writer. UNIX has a hierarchical file structure. Users can add or delete file directories at will and then "position" themselves at different locations in the resulting hierarchy to make it easier to manipulate files in the surrounding area.

The command interpreter of the operating system interface (called the "shell") can take its input from a file, which means that it is possible to put frequently used sequences of commands into a file and invoke that file (by typing its name), thereby executing the command strings. Because the shell includes facilities for passing arguments, for iterations, and for conditional operations, these "shell programs" can be quite proficient, essentially calling upon all system resources (including the editors) as subroutines.

By means of a communication channel, the output from one program can easily be directed to the input of another, allowing a sequence of programming modules to be strung together to do some task that in other systems would have to be done by a special purpose program.

UNIX does not provide special purpose programs, instead it attempts to provide a set of basic software tools that can be strung together in flexible ways using I/O redirection, communication channels, and shell programs.

The UNIX operating system is not covered by this manual.

INTRODUCTION

Several support systems are provided to enhance the performance of application packages running with RT-11, TSX or TSX-Plus on Spectrum computers.

3.1 TSX-NET

TSX-NET is a systems communications software package which allows communication and ease of access when using any combination of TSX and RT-11 systems with either DL-type serial lines or DZ-type multiplexer lines. The TSX-NET facilities permit a terminal to be connected to a remote system without affecting other data processing activities on the remote system.

Using TSX-NET, utility programs such as PIP, DUP and DIR, and user written programs may gain access to and manipulate RT-11 structured files on remote TSX or RT-11 systems that are connected via dial-up or dedicated RS-232 communication lines.

TSX-NET also features the following capabilities:

Provides transparent sharing of remotely located files and devices.

Incorporates a comprehensive data checking scheme to preclude the possibility of erroneous data transfers, and keeps statistics of these transfers to permit ongoing monitoring of data line integrity.

Allows one line printer to support several CPUs at a single site, thus reducing hardware investment.

Permits several TSX-Plus lines simultaneous shared access to remote device(s).

Contains a facility to software-connect a local terminal to a TSX time-sharing line on a remote system.

Permits backup of Winchester type media to industry standard removable devices installed on another machine.

3.2 RTSORT

RTSORT is a super-fast sort for systems running the RT-11, TSX or TSX-Plus operating systems.

It is designed to sort large disk files rapidly and uses the 'Replacement Selection' algorithm. Processing time is over-lapped with the I/O time to the fullest extent and buffer sizes are dynamically determined based on available memory space.

Typically RTSORT requires about 1.1 times the input file space to store temporary sort files on disk and requires a minimum of 16K of user space.

This assembly language sort handles both fixed and variable length records. The user can specify up to 16 sort keys, including signed and unsigned binary integers, single and double-precision floating-point numbers, ASCII character strings, dates and signed DIBOL-11 (or DBL) numbers.

3.3 VIRTUAL UNIT (VU) System

VU SATURN is a virtual unit system that improves disk organisation, and allows a physical disk to be divided into subfile directories.

VU SATURN is compatible with RT-11, TSX, and TSX-Plus. RT-11 disk files are created and function as virtual devices with their own directories. When a disk file is assigned to a particular virtual unit, the unit is treated like any other random access device.

VU SATURN includes an RT-11 device handler and a virtual unit assignment program. The handler enables eight virtual units (VU0: thru VU7:) to be active at any one time, and the assignment program allows virtual units to be assigned to different subdirectory files.

The specific features of VU SATURN are as follows:

Disk organisation is enhanced as programs, sources, and data files are easily segregated.

Different files with the same name can reside on one physical disk but in different virtual units.

Backup of specific data is easier. If all data is on a VU, the file assigned to the VU is copied.

Virtual units can be nested up to a maximum depth (7) possible.

Management and organisation of data is improved, directories are shortened, directory access times are improved, and system security on time-shared systems is enhanced.

On TSX-PLUS systems, the VU handler may be optionally generated to support 8 virtual units per line.

INTRODUCTION

The standard language options for Spectrum Eleven computers are normally selected from BASIC, COBOL, FORTRAN-IV and PASCAL. However, this list is not exhaustive, and Spectrums do support most available languages. If further information is required, please contact Webster Electronics.

4.1 SINGLE-USER BASIC-11 (V.02) Interpreter

BASIC-11 is an easy-to-learn, interactive programming development language which uses simple English words, understandable abbreviations, and the familiar symbols for mathematical and logical operations. In addition BASIC-11 also includes the capabilities necessary for file management, matrix manipulation and editing.

BASIC-11 is available as an option on RT-11 operating systems and requires at least 8K words of memory to run in systems with more than 8K words of memory. Any RT-11 supported device, including floppy and hard disks, magnetic tape and card reader can be accessed.

BASIC-11 may either execute statements as they are entered or execute whole files on storage media.

Other features include a complete set of string manipulation functions along with a dynamic allocation of string storage, PRINT-USING statement for output formatting, user defined functions, CALL statement for linkage assembly language subroutines, and graphics and laboratory peripherals support.

4.2 MULTI-USER BASIC-11 (V.02) Interpreter

MU BASIC-11 is the multi-user BASIC-11 system which supports from 1 to 8 users simultaneously and runs under the RT-11 foreground/background or XM monitor on systems with at least 28K words of memory. Users can access all devices supported by RT-11.

MU BASIC-11 provides a scheduling supervisor for multiple users and the system provides a log-on procedure and file protection as options.

All the features of MU BASIC-11, including statements, commands, functions, and immediate mode execution, are available to all users.

4.3 RJ-11 COBOL

COBOL is a locally supported product recommended for educational systems and small applications. It provides powerful record formatting, data manipulation and file handling capabilities, that are important for efficient data processing applications.

COBOL can be run under the RSTS/E and RSX-11M operating systems and can accept source program input from cards, console terminals, and disks, including input from source text library files stored on disks.

COBOL utilizes the Record Management Services (RMS) to implement user file handling. In addition COBOL programs can also create and/or read ANSI standard format magnetic tape files if magnetic tape sub-systems are configured in the system hardware.

Features include indexed file organisation, the 'SORT' verb, 'STRING/UNSTRING', and data editing facilities.

4.4 COBOL-Plus

RT-11 and TSX users have the option of COBOL-programming with COBOL-Plus, a full ANSI-'74 Standard COBOL compiler and run-time system.

Designed to run programs much larger than physical memory, the COBOL-Plus compiler allows automatic virtual memory program segmentation by a "least-recently-used" technique to keep the most active set of segments in memory while swapping the others to a disk file. COBOL-Plus run-time routines for OPEN-CLOSE, input-output, and ISAM are also managed in this fashion.

COBOL-Plus also provides interactive screen-handling capabilities through the use of positional ACCEPT and DISPLAY statements. Powerful input-output facilities are provided by support of sequential, relative, and indexed sequential files. Indexed access files are dynamically self-organizing. Record locking facilities are provided to synchronize access to shared relative and indexed organization files. Disk sorts are performed by the highly efficient and field-proven RTSORT facility.

4.5 FORTRAN IV (V2.0) Compiler

The entire FORTRAN IV system is operational in 8K words under the RT-11 SJ, FB, or XM monitors. Various features of the FB and SJ monitors can be utilized by the System Subroutine Library (SYSLIB) which is a collection of FORTRAN-callable routines. SYSLIB also

provides various utility functions, a complete character string manipulation package, and 2-word integer support. SYSLIB allows an RT-11 FORTRAN user to write almost all application programs in FORTRAN with no assembly language coding.

FORTRAN IV is characterized by its extremely rapid compilation rates and efficiency in small memory environments.

Despite its small size requirements and high compilation rate, FORTRAN IV also provides a high level of automatic object program optimisation. The compiler performs redundant expression elimination, constant expression folding, branch structure optimisation, and several types of subscripting optimizations.

As FORTRAN IV has no statement ordering requirements, declarations can appear anywhere within the source program. Terminal format input (using the tab character to delimit field) makes program preparation easier.

Larger FORTRAN programs are allowed as FORTRAN IV can allocate array storage outside a program's logical address space. These are virtual arrays and can contain any data type but they may also require operating system support of memory management directives.

Instead of using temporary files to process source programs, the FORTRAN IV compiler performs all its activities in main memory. It reads the entire source program once, stores it in memory in a compacted format, and processes the compacted code in memory. Since a disk device is not used for temporary file operations, the compilation speed is significantly increased.

During compilation, FORTRAN IV performs ten categories of program optimisation. These are: compiled FORMAT statements, array vectoring, constant folding, constant subscript evaluation, unreachable code elimination, common subexpression elimination, peephole optimisations, branch optimisation for arithmetic and logical IF, register allocation, and loop optimisation.

4.6 PASCAL-2 Compiler

The PASCAL-2 compiler is a high-performance optimizing compiler which can operate with most major operating systems, and supports all capabilities of standard Pascal. The compiler requires 28K words of main memory, approximately 500 blocks of disk storage, and the EIS hardware option.

The PASCAL-2 system includes utility programs and packages to aid the applications programmer with development, maintenance, and documentation. These are: a source formatter, a program formatter, a cross-reference index generator, a procedural cross-reference index program, a collection of procedures and functions for programing with

dynamic character strings, a library of MACRO-11 definitions, and a facility for automatic paragraphing and layout of text.

PASCAL-2 has the capabilities to include procedure and function parameters, non-local GOTO statements, set types of up to 256 elements, and packed data structures. PASCAL-2 supports integer types from 1 to 16 bits and single-precision or double-precision real types. Identifiers may be of any length, with all characters significant.

Pascal includes a number of convenience extensions: an include directive for combining multiple source files in a compilation, flexible ordering of declaration sections, structured constant declarations, and a default "otherwise" clause in the"case" statement.

Operating in five phases, the compiler employs virtual memory techniques to allow compilation of very large programs. Optimizations include global register allocation, constant folding, dead code elimination, short-circuit evaluation, expression targeting, array index simplification, branch-tail merging, and common subexpression elimination.

4.7 C

C is a general-purpose programming language that is useful for writing operating systems or for writing major numerical, text-processing, and database programs. Although closely associated with the UNIX operating system, C can be used with all major operating systems

C is a relatively "low level" language in that it deals with the same sort of objects that most computers do, namely characters, numbers, and addresses. These may be combined and moved about with the usual arithmetic and logical operations implemented by actual machines. C provides no operations to deal directly with composite objects such as character strings, sets, lists, or arrays considered as a whole. C does not define any storage allocation facility other than static definition and the stack discipline provided by the local variables of functions, and it provides no input/output facilities. All of these higher-level mechanisms must be provided by explicitly-called functions.

Similarly, C offers only straightforward, single-thread control flow constructions: tests, loops, grouping, and subprograms, but not multiprogramming, parallel operations, synchronization or co-routines.

One of the most important features of the language is its ability to define structures, and to manipulate them by use of pointer variables. In addition, an increment/decrement operator is available

which allows flexible indexing through a structure or vector. Multi-dimensioned arrays are also available, and variables may be partitioned into bit fields.

4.8 MACRO-11

The MACRO-11 assembly language, which uses the machine instruction set, can be used for development of programs and/or subroutines.

MACRO-11 processes source programs written in the MACRO assembly language, and produces a relocatable object module and optional assembly listing.

The MACRO-11 assembler is a two-pass assembler which provides for:

Global symbols for linking separately assembled object programs which promotes modular program design.

Device and file name specifications for input and output files.

User-defined macros.

Comprehensive system macro library.

Program sectioning directives.

Conditional assembly directives.

Assembly and listing control functions at program and command string levels.

Alphabetized, formatted symbol table listing.

Default error listing on command output device.

A Cross Reference Table (CREF) symbol listing.

The MACRO assembler also features:

Global arithmetic, global assignment operator, global label operator, and default global declarations.

Multiple macro libraries with fast access structure.

Predefined (default) register definitions.

An indirect command file facility for controlling the assembly process.

INTRODUCTION

This chapter introduces the general applications software, which incorporates a database management system, screen editors, and word processing packages, currently available with Spectrum Eleven computers.

5.1 WP-SATURN

WP-SATURN is a complete word processing/list processing package which can run under RT-11, TSX, or TSX-Plus. In addition it can be used with the RSX-11 and RSTS operating systems. WP-SATURN uses standard ASCII source files, provides automatic backup of files, and has file merge facilities.

WP-SATURN provides efficient, user-friendly word processing. The screen editor is cursor controlled and provides display of operator actions. Editing capabilities are also powerful and versatile. A user may copy, cut, or delete text, move columns, search for a text string, and do global or selective replaces. The system supplies many automatic formatting functions, including hyphenation, pagination, table of contents, indexing, paragraph numbering, and reformatting.

The list processing function is powerful and easy to use. It is a prompted system, from the designing of the format of a list, to entering data in a list, or editing that data. A user can change the format of a list after the data is entered and then convert the data to the new format without re-entering any data. A selective or global search and replace for fields within a list is provided. Lists can also be merged with word processing documents. In addition a sort function gives the user the capability of sorting a list or selecting a sublist from it. Multiple sort, select, and exclusion keys can be used.

WP-SATURN also has a dictionary to detect spelling errors and allows correction to be made by the user at the point of detection. A user has the facility to expand the dictionary if desired.

5.2 TEXT-11

TEXT-11 is a complete Word Processing/List Processing System which can run under the RT-11, TSX-Plus, or RSTS/E Operating Systems. It provides all the facilities required to:

Create and edit document files on the computer system;

Display and/or print at any time an index of all current documents, together with associated information such as document size, date of creation and date last printed;

Delete documents which are no longer required, together with their associated index entries;

Print single or multiple copies of any document on a variety of supported printer types;

Print multiple copies of a 'form letter' or other model document which contains variable information extracted from the user's existing computer files. This important feature being List Processing.

Operator selection of TEXT-11 functions is simplified by means of an hierarchy of menus. Comprehensive guidance is provided on the screen each occasion keyboard input is required and through the selection of a reserved 'help' key.

TEXT-11 is written entirely in MACRO-11 to achieve maximum performance. Single user support is provided for RT-11 systems, whereas under RSTS/E and TSX-Plus, TEXT-11 allows concurrent word processing operations at multiple terminals, and permits the use of multiple printing devices for high volume applications.

5.3 SIMILE

SIMILE is a database management system that allows a user to create, maintain, and reference a database. SIMILE operates with the RT-11, TSX-Plus, RSX-11M or RSTS/E operating system.

SIMILE is designed to help a user translate the complexity of his various activities into computer based operations, by using English-like commands. In a relatively short time the user can start entering data, perform queries and reports, as well as continue to develop and expand a system using more complex SIMILE concepts. Once built, the system can be run by operators having no knowledge of its internals.

SIMILE has the following features:

Schema utilities where databases are interactively created and modified.

Data security in the form of a password system to restrict access to databases and screen-files.

Flexible data operations for entry, modification and displaying, are performed quickly and easily.

Record selection where complex selection criteria are easily defined, and where SIMILE rapidly accesses only those records which have qualified.

Arithmetic definition where items may be defined as arithmetic relationships between other database items.

Database hierarchy where items may be defined as "pointers" into another database, thus providing a two-level hierarchy.

Conditional branching where error-handling and processing-flow can be controlled and directed by the system-builder.

External interaction of data files, created by SIMILE, which can be accessed by other user programs.

A large number of applications are suitable for SIMILE's range and these can be grouped into the requirements of commercial, educational, industrial, and scientific systems.

5.4 VDT

VDT is a screen editor, designed and developed by Saturn Systems Inc., which uses the basic functions of the WP-SATURN word processing package, and allows for original program creation, and modification of existing text.

VDT provides all necesssary features for editing. WP-SATURN attributes such as rulers, automatic word-wrap, and page headings are not included.

VDT is particularly useful for program preparation as it is very easy and quick to use.

5.5 FASTEXT

FASTEXT is a versatile full screen text editor for use with systems running RT-11, TSX-Plus, RSX-11M or RSTS/E. It is designed to operate efficiently with low system overhead (9K words) and no temporary work files, in multiple terminal environments, to eliminate system slowdown and delay problems.

FASTEXT provides a rich set of capabilities for the programmer, and basic word processing features for documentation preparation.

Support is available for VT52, VT100 (or full emulators), and Televideo 950 terminals.

FASTEXT features the following capabilities:

Allows the creation, inspection and editing of standard ASCII format text files. As no special characters are inserted into the file by FASTEXT, all text generated can be output to standard printers, compilers, assemblers, etc.

Provides rapid and accurate editing anywhere in the file, displaying that segment of the file as it will appear in the final document.

Has automatic wordwrap, supports user-selectable margins and two types of tab stops (to support "comment" tabs for comments in structured languages such as Pascal and C), and allows these pre-selected margins and tab settings to be stored in a ruler file.

Provides complete cursor control, permitting the user to scroll up and down through the file and to move quickly to the top or bottom of the file with a single command.

Incorporates automatic text centering, overtype, text insert and delete, as well as the search and replace, and cut and paste facilities.

Has provision for inserting files within files at the user's option.

INTRODUCTION

Webster Electronics markets a range of application packages which, combined with the appropriate hardware, form turn-key systems for end users. Installation and training are part of the package and on-going system maintenance is provided by Webster's customer service division.

There are a number of other, highly specialised packages available which require special marketing skills and experience. In these areas, Webster Electronics combines with the software house to provide the 'system solution'.

The areas covered by these systems include education, administration, various types of office/practice management, and scientific analysis. As further applications software is made available for Spectrum computers, this list will be extended in future editions of the manual. On request, Webster Electronics Sales Department can provide additional information on all available software.

6.1 MONECS

MONECS is a student educational system which provides an effective means for students to gain an understanding of computers. Batch processing is used with either punch or mark sense card decks and programs may be written in any of the popular computer languages (FORTRAN, COBOL, BASIC, PASCAL and MLP).

Features of the system include maintenance facilities that allow programs to be added or deleted from disk and system parameters to be changed, a report routine which prints a report of the current status of the log file and the list of files on the program library, and a routine to list cards on the printer.

A MONECS program library allows three different types of entry, ie., permanent entries of the maintenance functions, system library entries which are the various compilers, and a user program library. In addition MONECS has a data file library, which normally resides on its own disk, and is accessed through its own independent directory.

MONECS can also run under the RT-11 operating system as a single job or in the foreground of the foreground/background monitor. It looks identical to a stand-alone system and does allow an easier switch between MONECS and RT-11 facilities such as interactive BASIC.

6.2 T-MONECS

T-MONECS (terminal oriented system) designed and developed by Webster Electronics, is an enhanced version of MONECS. It is a software trainer which consists of a unique integrated set of programs to provide convenient and economical computing facilities for students.

T-MONECS supports the customary languages and uses the RT-11 operating system running with TSX (or TSX-Plus). In a time-sharing environment, five terminals may be used when running with TSX, and ten terminals with TSX-Plus.

RT-11 MONECS can still be run independently of T-MONECS by means of a pre-selected option, and T-MONECS can be run with programs originally created on cards and then copied to disk. This interactive use of a computer optimizes the machine time available and allows a high degree of programming ability to be achieved in a relatively short time.

Additional features of the T-MONECS system include:

a virtual unit system to improve management and organization of data, shorten directories, and enhance system security;

a screen editor (VDT) which allows for original program creation, and modification of existing text;

all the features of the MONECS package which includes the user accounting facility;

the MONECS graphics package which is an optional extra..

T-MONECS is fully documented and includes a Student Guide written solely for the benefit of students.

6.3 SCHOLAR

SCHOLAR, is a comprehensive, low cost, schools' administration system which consists of five interrelated modules designed to cater for the accounting and recording needs of independent schools. These modules are:

Fees statements (debtors)
Accounts payable (creditors)
General Ledger
Student records
Payroll

SCHOLAR thus provides for the requirements of student records, reporting and financial accounting within a school, and facilitates reporting to external authorities. These comprehensive reporting facilities provide teachers, department heads and administrators with accurate information on an as-required basis. SCHOLAR can be modified to suit a school's specific needs and further application programs can be added to integrate with existing modules.

The modules are written in BASIC and operate in an on-line interactive mode. All input produces output in the form of well designed audit trails, and automatic error detection and file recovery is provided.

6.4 T.P.C.M

TPCM (True Product Control for Multiple users) is a fully integrated production control system which incorporates production planning, bill of materials, product explosions and implosions, job costing, purchase order management and material requirements planning, work in progress, and machinery and labour utilization.

TPCM can be implemented as a stand alone set of programs which serve as an aid to manufacture and production, or alternatively, TPCM can be the initial stage of implementation of a totally integrated management system (M.A.T.E.) incorporating accounting, inventory, general ledger, payroll, and production functions, utilising a common data base.

Each function within the total system has been designed to allow for gradual or full implementation of the modules which provides a system totally suited to a user's specific requirements.

TPCM is menu-driven and is controlled by a major options menu split by functional groups. Each functional group subsequently covers a group of similar related functions. All options are selected by a four-character control verb and global control allows a function to be selected from any menu in the system. The data base provides accurate, up-to-date details which can be accessed simultaneously by production, engineering, sales or accounting personnel.

TPC is available for single-user systems.

6.5 T.A.S.M.

TASM (True Accounting System for Multiple users) is a fully integrated accounting system used in commercial, manufacturing, trading and service organizations, where the information is derived during the production of normal business documents rather than complex accounting interactions.

Simplicity is the keynote to the operation of the system and TASM is designed to interact with the operator, using a simple question and answer process.

TASM is a growth product that can lead ultimately to a total Management Information System (M.A.T.E.) when coupled to the TPCM product control system.

Like TPCM, the features incorporate global control which allows easy access to the various system functions.

TAS is available for single user systems.

6.6 PRACTICE MANAGEMENT SYSTEM

The Practice Management System completely automates most management functions of a medical office. The system is customized to each installation's individual requirements in terms of storage capacity and information content, and the same basic system satisfies the needs of general practitioners, specialists, dentists, and orthodontists.

The main benefits of the Practice Management System include:

immediate billing and receipting;

month-end processing which includes statements of account, aged balances, overdue accounts listing, and various reports and summaries;

storage and retrieval of a patient's records;

Special features are:

a menu-driven system where options are displayed at every processing stage;

patient records are accessed via surname and initials rather than via an artificial code;

commands are powerful and easy to follow; single-keystroke commands are used wherever possible;

reports may be printed or displayed on the screen; entries are alphabetic by surname without the need for a prior sorting operation.

6.7 MANAGEMENT COMPUTING SYSTEMS

These comprehensive commercial accounting systems cover a wide range of services designed for small business management. They are offered with full maintenance, installation, and operator training to clients.

In addition to the standard commercial packages summarised below, a number of specialised program sets have been written for different applications. These include such items as:

Real Estate
Job Control
Work-in-Progress
Asset Control
Mailing List
Price listing

Debtors

Interactive open-item system to trial balance. Instant comprehensive debtor enquiry, sales reporting, statement printing and full banking facility.

Stock Control

Full inventory control with multi-warehouse facility, profit and sales reporting and full control over stock reserved and backordered. Management-oriented summary reporting by category with optional foreign currency and work-in-progress modules.

Invoicing/Order Entry

On-line order entry provides high volume invoicing with comprehensive sales and representative commission analysis plus sales tax summary. Invoice line items can be backordered, reserved, forward-dated or supplied from any of multiple warehouses.

Creditors

Interactive system to Trial balance including details of entries up to 9 months ahead. Enquiry, remittance and cheque print plus comprehensive reporting and optional link to the General Ledger.

General Ledger

Enables overall system management reporting. Includes 12 months budgets and the ability to modify formats to different requirements. Trial balance, Profit and Loss and Balance Sheet reports with a comprehensive Chart of Accounts.

Payroll

Optional cost centre analysis if linked with General Ledger programs. Full facilities for a range of deductions, allowances and hourly rates, including payslips, paysummary, coinage analysis, enquiry and Group Certificate print.

Word Processing

Comprehensive word processing with list, dictionary and calculating facilities. Designed for memos, letters, label print, program development and technical documentation.

6.8 COMMERCIAL BUSINESS SYSTEMS

This series of commercial business systems have been designed to be stand alone or integrated depending on user requirements. The systems available are supplied in package form and cover the following application areas:

- Invoicing/Stock System
- Investment Register
- Professional Time Recording/Job Costing
- Debtors and Sales Analysis
- General Ledger
- Asset Register
- Payroll System
- Hire Purchase System
- Creditors System
- Trust Account System
- Stress - Information Storage and Retrieval Package
- Strain - Report Generation Package

The systems, designed for the small to medium sized businesses, are menu-driven to provide simple, easy to use operation. They are comprehensively documented and require minimal expertise from the user.

A brief summary of each system follows:

Invoicing System

The Invoicing System accepts details of goods sold and goods returned, producing from them Invoices and Credit Notes. It may communicate with the Stock and Debtors Systems, to provide complete Inventory and Accounts Receivable control. Invoice details (extended or not), credit details (extended or not), and cash sales details are input to the system interactively, and processed in batches.

Stock System

The Stock System provides complete inventory control for a company with multiple stock storage locations. Stock movements, to include inwards goods and stock adjustments, are input directly to the Stock System and stock moving out of inventory is obtained from sales details input to the Invoicing System.

Investment Register

The Investment Register provides control over monies invested with an investment company. The system is date driven on a user defined cyclic basis and provides for the recording of money invested by an individual investor, the calculation of interest due on a daily basis, the payment of that interest, and the production of various reports.

Time Recording and Job Costing

This system is designed to cater for the time and cost charging and analysis functions of professional firms, and to cater for time and materials costing for workshops and factories. The system is date driven on a monthly and end of financial year basis such that analysis is on a current month and year-to-date basis and job costings are on a current month and to-date basis.

Debtors and Sales Analysis System

The Debtors System is designed to allow flexibility and run time choice for the user in the selection and layout of reports and in the maintenance of debtors records. The Hire Purchase and Time & Cost Systems use the one debtors masterfile of names and addresses.
The Sales Analysis ledger can be built for each Debtors Ledger. The design of the ledger is determined by the user as well as the print layouts.

General Ledger

The system caters for both annual and monthly ledgers. Each ledger record within a given ledger contains only those fields that are specified when the ledger is built. Any ledger may have a fixed asset and depreciation ledger which can be run independently of the general ledger or in conjunction with the general ledger, such that depreciation transactions are generated for posting to the general ledger.

Asset Register

Each general ledger may have a fixed assets and depreciation ledger associated with it. Depreciation charges may be posted in user specified proportions to nominated account codes in the general ledger. The ledger is date driven and may be linked to the associated general ledger. A fixed assets and depreciation ledger may exist without being linked to a general ledger.

Payroll System

The Payroll System has been designed to cater for different payrolls of various sizes. Ease of operation, speed and flexibility are the main features of the system. Reports include two payslips for each employer, costing report, full YTD reporting for the Tax year, current period earnings report, allowance and deduction analysis report, and pays to be banked.

Hire Purchase System

The system is designed to provide the user with the information needed to control credit and cash input for goods on hire purchase. It is primarily intended to be used as a record keeping system reporting on status of hire purchase agreements. The system caters for five types of repayment periods (weekly, monthly, etc), and all five types may be combined in one ledger.

Creditors System

The Creditors System is designed as an open item system and can handle manual payments where payment is made outside the system, in addition to payment on account where no specific individual transaction is paid. This system may be interfaced to the General Ledger System for analysis purposes. Creditor transactions may be analysed in either of two ways: by private ledger code and/or by product code. This requires the existence of a monthly general ledger for each analysis selected.

Trust Account System

This system enables chartered accountants or solicitors to control and record clients' trust and investment accounts. Various reports produced by the system include a trial balance, investors register, and investment register.

Information Storage and Retrieval Package

This system is intended to provide the user with a flexible multi-purpose record-keeping system. Possible applications of this system include:

Membership system;
Diary/reminder system;
Office record keeping;
Taxation reminder system;
Mailing lists.

The user has the ability to specify what data is to be stored and in what format. Data can be updated on either an individual record basis or in user-selected groups. The user can retrieve data records according to selection criteria, and these may be reported on in record key sequence or according to user-defined sort keys.

Report Generation Package

This is a read-only program which is used to retrieve records which meet user-defined criteria and to print data from these records according to a user-defined print format in a user-selected reporting sequence.

6.9 PIG MANAGEMENT SYSTEM

The Pig Management System (P.M.S.) offers a fully comprehensive pig management program which includes sales and cost analysis, production and forecasting, budgetary control, standardised accounting, and least cost rations.

P.M.S. is designed to assist in the breeding of superior stock and gathers information on each animal, which includes details of litters and their subsequent survival capacity. This in turn produces information on where genetic defects occur, when to retire an animal from breeding, and who the superior breeders are, at any given point in time.

P.M.S. controls the accounting and practical side of pig farming and offers reporting facilities on a day to day basis which will ultimately increase profits. This includes such items as feeding schedules and costs, and the status of an animal in terms of its readiness for market. Past sales analysis figures can be obtained from the system and the results when related to the breeding program, help forecast the sales and production for any given time period.

6.10 COMPUTER AIDED PLASTIC MOLDING SYSTEM

The Moldflow System is a scientific method of plastic mould and part design which can be applied to complex shapes.

Moldflow is a series of computer programs which analyse the flow of hot plastic into a cold mould, by considering both heat transfer and fluid flow, to predict: pressure, temperature, stress level, cooling time and other factors, throughout the mould. The underlying concept of the Moldflow system is to design the flow path to control the flow of plastic within the cavity. This method produces Moldflow parts which have lower stress, reduced warpage and cracking, shorter cycle times, reduced overpack, lower weight, and smaller runner systems.

Moldflow is available for use under a world wide time-sharing system or as a complete turn-key minicomputer package. This package, which includes the Moldflow software, a Spectrum 11 GC computer, four video terminals, a digitiser, and graphics terminal, is further enhanced by a communications link to the time-sharing system which provides the latest technical developments.

Where access to graphics is available, design practicability of individual experiments can be evaluated and potential problem parts re-designed before any actual moulding takes place.

A unique property data bank is accessed by the system. This bank, which has been built up with the co-operation of the world's major international suppliers of plastic polymers, enables users to select the best grade of material for a particular application, and provide on screen the most suitable mould design. The system can also position weld lines in the least sensitive areas.

In addition to its design capabilities, the Moldflow system is used as a trouble-shooting device to analyse existing moulds simulated on screen.

6.11 MEDIPAC

Medipac is an economical Practice Management system designed to meet all the up-to-date requirements of a solo or group medical practice.

The Medipac system ensures accurate recording and presentation of all information required for the efficient management of a practice, and provides the following features:

prompt collection of fees with instant memoranda of cash/cheque receipt at time of visit;

recording of all cash and cheques received, producing a bank deposit slip;

statement of a patient's account showing previous and current balances owing;

month-end processing which includes statements, overdue accounts and trial balance reports;

an efficient appointment booking system;

maintenance of individual patient, family, and company accounts;

monthly analysis of services rendered by service type and doctor

budget option for payment of account.

Medipac is easily understood, simple to operate, and can be installed immediately with no disruption to normal work flow.

6.12 C.A.R.E.

CARE provides a control and review mechanism of business transactions by Management. It is a simple, yet effective means of introducing management control to the visible aspects of an administration.

CARE is entirely menu driven and is designed and developed with the first-time or non EDP-oriented user in mind. All system functions are password protected, passwords being set by the system administrator to associate function privilege with user status.

CARE is built around a central user defined database, the contents of which form the basis upon which business transactions are checked at time of entry. These transactions may be maintained interactively at any time and can take a variety of forms. For example:

Request for proposal;
Product evaluation;
Proposal evaluation;
Union claims;
Forms of litigation;
Budget preparation;
Staff recruitment;
Workers Compensation claims.

Progress of the transaction is then reported back to CARE as the evaluation/decision making process advances. CARE provides facilities to monitor the status of specific transactions on demand, to report management and exception information on specific transactions, and to obtain throughput on a regular and ad hoc basis. A complete audit trail for all transaction activity is maintained by the system.

6.13 SIMPL

The SIMPL Corporate Planning System, when integrated with the General Ledger system, provides a complete management information system.

SIMPL can compare actual results to budget for the current month and on a year-to-date basis. It can provide re-forecasts each month on latest estimates for the remainder of the year, to highlight any potential problems for year-end financial statements or future bank balances, and provide exception reports to detail items which have varied significantly from budget.

SIMPL also features cost estimating models in which the activities involved in providing goods or services to the market can be simulated. This provides a total estimate for cost-of-sales and allows for recalculation on variations before quoting future prices.

6.14 LEGAS

LEGAS, Legal Accounting software, is a fully integrated legal accounting system which looks after all aspects of accounting in a legal office. The main features of LEGAS can be attributed to:

Trust accounting;
Disbursements ledger;
Debtors ledger;
Time recording and billing;
Private ledger including creditors ledger;
Contributory mortgage accounting;
Short term deposit accounting;
Multiple branch accounting;
Automatic archiving;
Cheque and Receipt printing.

Many benefits are gained from using LEGAS and these include:

the use of the 'open-item' accounting method;

a total control being maintained over charges, relative to the subject matter.

a management analysis to show comparison of value of work with fees charged, to highlight discrepancies of these charges within and between departments, and to summarize outstanding fees and disbursements for each Solicitor.

total billings by client that enables the firm's major clients to be identified and their level of business with the firm analysed.

full matter enquiry facility.

a trial balance listing of trust ledger available in three formats.

a general ledger transaction listing which doubles as a simple Profit & Loss Account and Balance Sheet.

In addition LEGAS provides a record of details on each client, partner or associate, and each matter. Complete maintenance of these master files ensures availability of accurate details, while efficient recording methods allow details of all transactions to be displayed on the screen or printed out as hard copy if required.

Password security ensures restricted access to items of a confidential nature and this includes entry into the LEGAS system, entry into file maintenance routines, and enquiry entry into the private ledger. Rigorous controls are implemented throughout the system and include a complete audit trail of the day's transactions.

LEGAS is available as a basic system and depending on user requirements can include the various additional modules. The integration of these modules puts no restriction on the different accounting periods as LEGAS operates on four different dates at any one time.

6.15 PROPERTY MANAGEMENT SYSTEM

The Property Management System is a completely interactive on-line system that is structured to handle the following tasks as a single requirement or as a combined operation.

Residential Property Management;
Commercial/Industrial Property Management;
Strata Management;
Holiday Letting and Booking Management.

These modules are served by three master files, Owners, Tenants, and Property, and the system is designed so that a transaction once entered, will automatically post to and update all the relevant files.

The various functions performed by the system include:

Issue of rental and other receipts, details of which are posted to the appropriate files.

The printing of the cash book which sections the cash receipts, cash payments, and the general journal. This is followed by the bank deposit slip with full details of cash and cheque credits and individual details of cheques.

Drawing of cheques for miscellaneous payments against owner's accounts, with password protection available for amounts over a nominated limit.

Owner statements which detail tenant's payments, and other credits and payments against owner's account. This is summarised to produce the total amount to be disbursed to the owner and a cheque is subsequently printed.

Printouts of owners and tenants ledgers showing complete details, to conform with the appropriate state legislation.

Various reports, which may be printed on demand, include a trial balance, statistical report in current management, arrears listing, and master file listings in several formats.

Controls are built into the system and include a complete audit trail on all transactions.

The additional modules which may be integrated, provide details and reports on commercial properties, and booking facilities for holiday accommodation.

6.16 SATURN-CALC

SATURN-CALC is an electronic spread sheet and calculator giving the user the capability to prepare sales forecasts and financial statements.

SATURN-CALC is capable of producing a spread sheet of up to 52 columns by 999 rows into areas called cells, with a means of scrolling the worksheet to different areas if it is too large to fit on the screen. In addition, a split screen feature allows viewing of two different sections of the worksheet at the same time.

SATURN-CALC accepts values such as labels, numbers and formulae, and prompts the user to enter the required command.

Entering of data is simplified as SATURN-CALC operates in insert mode and characters already typed in move to the right to make room for added characters. The various formulae available are expressed by a symbol notation and commands are either a one- or two-character code preceded by a slash character. In addition, SATURN-CALC uses global commands which affect all cells within a worksheet.

Other features of SATURN-CALC include:

Special commands to format and provide fixed point, scientific, percentage, or dollar notation for individual cells.

Functions which are common mathematical formulae or values and are represented by a function word.

SATURN-CALC is very simple to operate and all movement around the screen is controlled by the arrow (cursor) keys. These keys move the cursor without affecting the text on the line.

Many of the commands provide great flexibility within a document and allow a user to move/insert columns and rows, replicate cells, lock/unlock cells, and clear all cells. Commands that can greatly change a worksheet require verification to a system prompt.

INTRODUCTION

This chapter discusses the basic software procedures that should be carried out when using the RT-11 operating system with a Spectrum Eleven computer.

Reference is made to several chapters in this software manual and in the Spectrum Eleven Hardware Manual. It is recommended that these be read in conjunction with this chapter. Where applicable, reference codes are included in the text.

SPECTRUM ELEVEN HARDWARE MANUAL
(abbreviated as HM))

Section 1 - Introduction to the Spectrum Eleven (**HM1**)

Section 3 - Operations:

'Device' Operator Controls and Procedures (**HM3**)
Bootstrap Procedures (**HM3/5**)
Problem Checklist (**HM3/6**)

SPECTRUM ELEVEN RT-11 SOFTWARE MANUAL
(abbreviated as SM)

Section 2 - RT-11 Installation Guide:

System Disks
RT-11 System Conventions (**SM2/3**)

Section 3 - Run-Time RT-11 User Guide (**SM3**)

Section 5 - Spectrum RT-11 Operator Utilities (**SM5**)

1.1 SYSTEM CONFIGURATION

Before using the Spectrum Eleven computer the hardware configuration, including the identification codes for storage volumes and peripheral devices, must be established (**HM1**)

As an example Spectrum model **SS23GCM3** consists of:

SS23	An LSI-11/23 processor	
G	A single 30 megabyte fixed Winchester disk	(DM0:)
C	A single 1.26 megabyte floppy disk	(SF0:)
M	A single 40 megabyte magtape	(MT0:)
3	128 kilobyte MOS memory	

In this configuration DM0: is the volume that would normally be used for all system/application program storage. To establish the device codes for the peripherals on your system, refer Table 3-2 (**SM2/3**).

1.2 STARTING THE RT-11 COMPUTER SYSTEM

On a Spectrum system, the operator controls (**HM3**) are used only to start the system. If problems are encountered , refer **HM3/6**. Once the system has been started, interraction with the RT-11 computer system occurs through the terminal.

Starting the system involves turning on the computer along with the various hardware devices, and loading the appropriate software components into computer memory. Computer memory is where system information and data are temporarily loaded and stored for use during the various system operations. To use the system, computer memory must contain the RT-11 operating system and in particular the RT-11 monitor program. The process of transferring the RT-11 monitor from the system disk to memory is called bootstrapping the system (**HM3/5**).

1.3 INTERACTING WITH THE RT-11 COMPUTER SYSTEM

On completion of a successful bootstrap, RT-11 is installed as the operating system, and a monitor prompt (denoted by a dot) displays at the left margin of the screen on the console terminal. Since the console terminal can perform both input and output operations, it is used to interface between the system and the user. Commands are entered from the terminal keyboard and the computer prints information and messages on the terminal screen. In each system there is at least one terminal through which all system/user interraction takes place. This is called the console terminal (TT:). If a system has more than one terminal, one of them is still designated the console terminal.

On Spectrum configurations, the console terminal has a keyboard and a video screen. The instruction manual, supplied by the manufacturer of the terminal, will give full details on the use of the various control keys.

The presence of the monitor prompt means that the computer is now waiting for a command to be typed. The RT-11 monitor command language is described in the next section (**SM3**). The kinds of command operations usually performed immediately after the monitor is bootstrapped are those that set up initial conditions such as current date and time of day, and those that initialize and prepare the system for future operations.

The following comments refer to the configuration example used in paragraph 1.1.

During bootstrap DM0: is automatically assigned as the system device (SY:) and as the default storage device (DK:). Refer **SM2/3.** In this example the logical device codes SY: or DK: can be used for the system volume in addition to the standard device name DM0: Most keyboard monitor (KMON) commands are applicable to the default storage device (only the R command requires that the file be specified on the system volume). Since the system volume is initially the default storage volume for all operations, it is not mandatory to use a device name for the system volume.

Lets assume that the floppy disk device SF0: is to be the default storage volume. Before any processing is done a command must be entered to assign it to DK:

```
ASS SF0 DK
```

For verification, the SHOW command displays all assignments. As SF0: is now the default storage device, every time a command is entered it will apply to the floppy disk and not the Winchester disk.

As logical device assignments are temporary, when a device assignment is to remain in effect, it must be assigned each time the system is bootstrapped. It is common practice to install these installation-standard commands in a system startup command file. This saves having to enter the established system startup commands each time the system is bootstrapped. To find out what these commands are, type in the following command:

```
TYPE STARTS.COM     (if a single-job monitor)

     or

TYPE STARTF.COM     (if a foreground/background monitor)

     or

TYPE STARTX.COM     (if an extended memory monitor)
```

The commands included in the .COM file on your system are all required to be activated each time the system is bootstrapped so that correct processing can take place.

1.4 STORAGE VOLUMES

At this point a user should establish the state of the storage volume in terms of system and data file content, and amount of free space available.

The DIR (Directory) command is entered. Bear in mind that if DIR only is entered, a directory of the default device is displayed. As

stated, the system volume SY: which is the bootstrap device serves as the default storage device for all system operations unless DK: is assigned to another volume.

To establish the files and free space available on the system disk enter:

DIR/FULL (system volume is the default device)

To establish the files and free space on the work disk/unit enter:

DIR/FULL DK:

The Directory command displays the file name, the number of blocks of space taken up by that file, and the file's creation date. Although there may appear to be sufficient space left on the volume, it may be fragmented. To consolidate all empty space at the end of a volume, the SQUEEZE command is used. This command should never be used until a backup of the disk has been taken. If the files on a floppy disk are to be squeezed, the command is:

SQU SF0:

If the floppy disk (SF0:) had been assigned as DK: this command could be entered:

SQU DK:

When all files on a volume are no longer required, the volume is initialized. For example, to clear the directory of SF0: the command is:

INIT SF0:

A new (unused volume) should always be initialized before it is first used.

CAUTION Take heed of the warning message and ensure that the correct volume is being initialized.

1.5 SYSTEM SECURITY

At least once during a session on the computer, a backup copy should be made of the work volume. As the system volume is usually copied from the Distribution disk, a permanent security copy should always be available.

Backup can be done in several ways; either with the RT-11 COPY command or by using one of the Webster utilities (refer **SM5**). If using COPY for a disk copy, ensure that the output disk is first initialized with the INITIALIZE command.

Lets assume that five particular files from DM0: are copied to SF0: using the COPY command. Each time these files are copied the latest version overwrites the previous version so that five files are still present on SF0: One obsolete file is then deleted from DM0: and only four files are copied to SF0: However the obsolete file is still present on SF0: from the last copying session. This file will remain on the floppy disk unless deleted by the DELETE command. To overcome this situation it is preferable to initialize the volume in SF0: each time before using the COPY command. A recommended practice is to copy files to alternate disks so that the previous backup is always available.

1.6 RUNNING AN APPLICATION PROGRAM

In some installations, where a single application system is run, the application system program is included in the system startup command file. This means that the beginning of the job is displayed on the screen (usually in the form of a Menu) and no action at all is required by the user.

However, where the user must instigate the startup of a particular application, the R (Run) command is used with the application system's .SAV filename. Ie., to run the floppy disk utility FUTIL, the command is:

R FUTIL (this invokes FUTIL.SAV from the system disk).

A common feature of application systems is the inclusion of an option to return to RT-11. When this option is selected the system returns to the monitor prompt and allows the user to perform other functions.

1.7 ABORTING AN APPLICATION PROGRAM

To stop any currently running program, enter CTRL/C twice in succession. Note that the CTRL key is held down while C is depressed.

INTRODUCTION

This chapter specifies the files available on the Spectrum distribution disks and discusses various disk and file maintenance functions available with RT-11.

2.1 RT-11 LICENCES

There are two forms of software licence available for RT-11 systems:

An RT-11 run-time licence system which contains enough files to run application programs and perform file maintenance but does not enable the user to assemble and link new programs.

A full RT-11 licence system, which in addition to the files contained in the run-time system, contains all files necessary to assemble and link new programs.

These files are described in the next paragraph.

Also included on the standard distribution disk are one or more Spectrum support utilities. The selection of these is dependent on the storage media used on the system. Full instructions on the use of these utilities are documented in Sections 5 and 6.

FUTIL.SAV
BACKUP.SAV
BAKFLP.SAV
SAVRES.SAV
RANDY.SAV
WUTIL.SAV

In addition, further device handlers, which are not standard, are available separately and can also include any special customized handlers for particular applications.

Finally, a file ADVERT.LST, which is not standard, may be included. It contains a Webster Electronics title display which appears on the screen after bootstrap.

2.2 RT-11 SYSTEM FILE DESCRIPTIONS

File names which are highlighted by an asterisk are included on the RT-11 run-time distribution disk.

2.2.1 Indirect Startup Command File - (.COM)

RT-11 provides the ability to execute RT-11 commands from a file. Refer to next chapter 'RT-11 System Conventions' for details of Indirect files.

STARTS.COM*	Allows for automatic command file execution at start up for a single job monitor.
STARTF.COM*	Allows for automatic command file execution at start up for a foreground/background monitor.
STARTX.COM	Allows for automatic command file execution at start up for an extended memory monitor.

2.2.2 RT-11 System Files and Device Handlers - (.SYS)

MONITORS:

An RT-11 system always contains a monitor which can be one of the following:

RT11SJ.SYS*	The single-job (SJ) monitor provides an environment suitable for developing simple FORTRAN or BASIC applications. It supports all hardware devices except MMU (Memory Management Unit), many programmed requests, and all the utility programs. This monitor is for single users not requiring FB operation or the additional FB features.
RT11FB.SYS	This monitor allows two programs to operate; a foreground program and a background program. The real-time function is accomplished in the foreground. Functions that do not have critical response time requirements, such as program development, are accomplished in the background and this operates whenever the foreground program cannot run.
RT11S2.SYS	A single job monitor for two terminal support.
RT11S6.SYS	A single job monitor for six terminal support.
RT11F2.SYS	A foreground/background monitor for two terminals.

RT11F6.SYS*	A foreground/background monitor for six terminals. This monitor is included with the SJ monitor for run-time systems so that MU BASIC-11 may be run.
RT11XM.SYS	An extended memory monitor which provides means for a program to access up to 248Kb of physical memory. This monitor is rarely used.

SWAP FILE:

SWAP.SYS*	This file is used to hold certain parts of the operating system in order to temporarily free the memory that would otherwise be occupied.

HANDLERS:

A handler exists for every device that the system supports:

BA.SYS*	Pseudo device for RT-11 Batch system.
CR.SYS*	Card Reader
DD.SYS*	TU58 DECtape.
DM.SYS*	RK06/RK07 Winchester disk.
LP.SYS*	Parallel Line Printer.
LS.SYS*	Serial Line Printer.
MQ.SYS	Message queue pseudo-device for inter-job communications under the FB monitor.
MT.SYS*	Magnetic tape.
NL.SYS*	Null device.
PC.SYS*	Paper Tape combined Reader Punch.
RK.SYS*	RK05 Disk Cartridge.
SF.SYS*	Spectrum Floppy disk handler for all formats including 2D, 1D, 1S, DX, and IBM.
SM.SYS*	Spectrum Magtape
TT.SYS*	The single job monitor requires this handler in order to communicate with the terminal.

2.2.3 Memory Image Files - (.SAV)

A memory image file is a picture of what memory looks like after a program is loaded. A memory image file can run in an SJ environment or as a background job under the FB monitor.

BATCH.SAV	BATCH is a complete job control subsystem that provides batch-mode processing of user jobs in both the SJ and FB environments. BATCH processes job streams in the background partition, allowing real-time jobs or other user jobs to run in the foreground.
BINCOM.SAV	A binary file comparison utility that helps locate the changes made in source files.
CREF.SAV	A cross-reference utility used in MACRO-11.
DIR.SAV*	A utility used to list the file directory for file-structured devices. DIR allows directory listings sorted by file name, file type, date, size, or position.
DUMP.SAV	Allows the contents of a file to be printed on the line printer, listed on the terminal or written to a file, in octal words, octal bytes, ASCII characters, or Radix-50 characters. DUMP is particularly useful for examining directories and files that contain binary data.
DUP.SAV*	A disk utility that performs functions such as creating files on file-structured RT-11 devices, extending files on certain file-structured devices, and it can compress, image copy, initialize, or boot RT-11 file-structured devices.
EDIT.SAV*	A text editor used to create and modify ASCII text files. Both character and line oriented commands are included, along with provisions for command interaction, editing macros, and file manipulation.
FILEX.SAV*	A utility that allows transfer of RT-11 files to and from some other operating system environments. It does not support magtapes, cassettes or double-density floppy disks.

FORMAT.SAV — A utility that allows the user to format RK05 disks.

HELP.SAV* — Allows a user to access useful information about keyboard commands.

K52.SAV* and KED.SAV*

These keypad editors are designed for use on terminals such as the VT100 and VT52 as they incorporate the additional function keypad keys. This allows a user to position a cursor anywhere in the text file and to make amendments with their effect immediately visible on the screen.

LIBR.SAV — A librarian that creates and maintains libraries of commonly used object module routines and assembly language macro definitions.

LINK.SAV — The linker converts the relocatable object modules produced by the assembler or optional compilers, into a format suitable for loading and execution.

MACRO.SAV — The MACRO assembler.

PAT.SAV — An object module patch program that performs minor modifications to files in object format (.OBJ files).

PATCH.SAV — Is used to make code modifications to any RT-11 file but cannot be used to modify files linked with the RT-11 V04 linker.

PIP.SAV* — A Peripheral Interchange program that allows transfer of files (ASCII or binary) between any RT-11 supported devices. It also allows the user to merge, rename, and delete files.

QUEMAN.SAV — A background job that processes command lines and file specifications which have been entered. It is a user interface for the QUEUE utility (QUEUE.REL)

RESORC.SAV* — The resource program examines the currently running RT-11 system and displays useful information about the status of the monitor and the system configuration.

SIPP.SAV — A save image patch program that can be used to patch files that have been created with RT-11 Version 4, and for pre Version 4 programs that are not overlaid.

SLP.SAV — A source file patch program that provides an easy way to make changes to source files.

SRCCOM.SAV — An ASCII file comparison program that helps locate the changes made in source files and outputs differences to a specified output device. SRCCOM can also create an input command file to SLP.

TECO.SAV* — A text editor used to create and modify ASCII files. This is a much more powerful editor for the advanced programmer, which includes scope support, pidgeon-hole Q registers, field replace, and complex command macro programmability.

2.2.4 Executable Foreground Program File - (.REL)

QUEUE.REL — A device queue manager that runs as part of the QUEUE package as a foreground or system job.

2.2.5 System Macro Library - (.SML)

SYSMAC.SML — The default macro library containing many useful macros.

2.2.6 Object Output File or Library File - (.OBJ)

ODT.OBJ — The on-line debugging technique utility which aids in interactively testing and correcting assembled and linked object programs.

SYSLIB.OBJ — The default subroutine library which contains useful subroutines which have been written in a language, eg., MACRO, FORTRAN, PASCAL, and have then been translated into .OBJ format.

2.2.7 TECO Files - (.TXT, .TEC, .TC, .TMP)

TECO has several system files which are included on the distribution disk for the full licence system.

README.TXT	This file contains the documentation text for all TECO related files.
INSERT.TEC	TECO macro for immediate mode editing with VT11.
LOCAL.TEC	TECO macro to renumber local symbol blocks in MACRO assembly language programs.
SORT.TEC	TECO macro for general purpose memory resident sorting.
SQU.TEC	TECO macro to compress other TECO macros.
VTEDIT.TEC	TECO macro for very powerful scope editor, VT52 or VT100.
TECO.TC	TECO macro used with VTEDIT.TEC
EDIT.TEC	TECO macro for convenient startup with VT11.
SEARCH.TEC	TECO macro for locating one or more strings in one or more files.
VEG.TEC	TECO macro for convenient startup with VT52.
TYPE.TEC	TECO macro for typing files (allows many useful options).
VT52.TEC	TECO macro for immediate mode editing with VT52.
TECF00.TMP	This file is used to hold the name of the file currently being edited by TECO.

2.3 SYSTEM DISK DESCRIPTION

Every RT-11 system requires some sort of bulk storage device to store the system software. This device, usually a disk drive, is called the system device and contains the RT-11 files collectively known as the system. The disk itself is referred to as the system disk and Spectrum machines use as the system device (refer paragraph 3.6), either a floppy disk, cartridge disk, or Winchester disk.

2.4 SYSTEM DISK CREATION

In case of system software failure, the system files are recovered from the master distribution disk.

Firstly INITIALIZE the target disk, and then COPY all files from the distribution disk using the /SYS option. Finally COPY/BOOT so that the bootstrap is installed on the new disk.

Take extreme care if using the COPY/DEVICE command with storage devices of different sizes. Refer paragraph 2.6.2 for instructions.

2.5 FILE MAINTENANCE

2.5.1 Floppy Disks:

It is a known factor that floppy disks deteriorate with use as the read and write heads actually press against the disk surface and this eventually wears. A good policy is to cycle floppy disks and assign them for use as backup disks after a period of use as work disks.

Used disks should be periodically checked with either the RT-11 command DIR/BAD or the Webster utility FUTIL. Of these FUTIL is the most useful in checking floppy disks for potential errors in that it can detect soft, ie., recoverable, errors. These soft errors indicate possible future disk problems.

File Recovery on Bad Disks

Once a hard (unrecoverable) error has occurred on a disk, the following steps should be taken to assess the damage and attempt to recover any lost information.

1. Perform a DIR/BAD/FILES on the disk to see what (if any) file has been affected. It may be that the bad area is on an unused portion of the disk.

2. Copy all files off that disk, except the one(s) that have the bad blocks, using the /EXCLUDE option of the COPY command. If no file has bad blocks then the information has been successfully recovered from the disk. For single floppy disk systems this copying operation is done with the COPY/WAIT command. If FUTIL is used instead of the above method, the whole disk including bad data is copied.

3. Attempt to copy the bad files with COPY/SLOWLY. This option reads the file one block at a time and may successfully read bad blocks. If this fails then use COPY/IGNORE to ignore errors. This will certainly read the file but the recovered bad block must be checked (the location will have been given by DIR/BAD/FILES).

Using Bad Disks

As a matter of course, after all information has been recovered from a bad disk, the disk should be reformatted (using FUTIL) as this may correct hard errors. If it does, the disk may then be used, although such floppy disks should be considered suspect and used for backup only.

Disks with bad blocks may be used after using the INIT/BAD command to re-initialize them as this writes .BAD files over the bad blocks.

2.5.2 Cartridge Disks:

Back up all files, that normally reside on the suspect removable disk, to another disk. The same RT-11 commands are used as with the floppy disks. Use the INIT/BAD command to reinitialize the suspect disk but this should be considered a temporary expedient and the disk should be reported to field service for checking and possible refurbishing.

2.5.3 Winchester Disks:

Use the DIR/BAD command to ascertain if and where any bad blocks are located. If possible back up the whole disk to either magtape or a series of floppy disks (using SAVRES) and report the state of the Winchester to field service.

2.6 RT-11 COMMAND CONSIDERATIONS

2.6.1 Copying Procedures

A copy procedure reproduces data without altering the original data.

Disk Copy — Copy is a facility to transfer files or volumes from one media to another.

File Copy — File copy will transfer a named file or files from any size volume to any size volume. It transfers the filename to the volume directory and it's associated data to the next free space on the output

volume. Certain options must be included in the command for system and protected files.

Boot Copy — If the volume created by a file copy is to be a system disk, then the BOOT information has to be copied into track 0 of that volume.

Device Copy — The COPY/DEV command will copy a complete volume to another volume of the same or different size.

FUTIL copy — This utility is for floppy disks only and is the same as the device copy with the addition of the error reports (soft and hard errors).

BAKFLP copy — The same as the file copy, plus security in the form of volume ID requests, and the facility of providing for file overflow from one storage volume of the same type to the next.

2.6.2 Volume Size Squeeze

As volumes vary in size from one storage device to another, care must be taken when COPY/DEVICE is executed. The RT-11 COPY/DEVICE operates on a block for block copy until the end of the smallest volume is reached. No warning is issued if the input volume still contains data not copied and the output volume is full. COPY/DEV from a smaller volume to a larger volume will therefore result in the exact image of the small volume on the large volume. Any copy under FUTIL or COPY/DEVICE between volumes of different size has to be 'squeezed' for directory update according to its space on that volume. Also, when copying from a large volume to a small volume, files may have to be first DELETED then SQUEEZED on the input volume in order that the total active area does not exceed the output volume size.

SQUEEZE is therefore a useful but very dangerous operation. The only way to ensure not losing a volume when SQUEEZE is executed, is to copy that volume to a backup volume before starting Squeeze.
SQUEEZE will rewrite the whole disk excluding the BOOT. All empty entries are moved to the end of that volume as free space. The volume size is re-entered in the directory area according to the file size and empty area on that volume but directories are still initialized according to the source volume's segments.

INTRODUCTION

This chapter, written for the benefit of users unfamiliar with computer systems, introduces some aspects and components of the RT-11 Operating System which require clarification. It is suggested that the subsequent Section is referenced when monitor commands are specified.

3.1 FILES

A file is an organized collection of related data. Each file on a disk (more properly a 'volume' as magnetic tape contains files as well) has a unique name consisting of a filename and filetype. The name is from one to six alphanumeric characters and the type (which is optional) is from one to three alphanumeric characters. No spaces or tabs are allowed in the file name or file type.

The names identify a file and distinguish it from all others on a 'volume' and the file type generally indicates the format or contents of a file. The type is often referred to as a file extension (ext).

Conventional file types are listed in Table 3-1.

TABLE 3-1

.BAC	BASIC compiled file, single precision
.BAD	Files with bad (unreadable) blocks
.BAK	Editor backup file
.BAS	BASIC source file
.BAT	BATCH command file
.BAX	BASIC compiled file, double precision
.COB	COBOL source file
.COM	Indirect command file
.DAT	BASIC or FORTRAN data file
.DIR	Directory listing file
.DOC	Documentation file
.FOR	FORTRAN source file
.LST	Listing file
.MAC	MACRO source file
.OBJ	MACRO or FORTRAN object output file or library file
.REL	Executable foreground program file
.RNO	RUNOFF format text file
.SAV	Executable background program file
.SML	System Macro Library
.SYS	RT-11 System files and device handlers
.TEC	TECO source file
.TMP	Temporary file
.TSX	TSX System file
.TXT	Text file

3.2 FILE PROTECTION PROCEDURES

The simplest way to protect a file is to make a copy of it on a second storage volume. There are several Webster Utilities that can do this job:

FUTIL or BACKUP for use with a floppy disk to floppy disk copy;

BAKFLP for a Winchester disk (selected file) to floppy disk copy

SAVRES for a Winchester disk to magtape/floppy disk copy.

In addition, the RT-11 COPY command can be used.

A further protection feature is the provision of a Write-Protect button on SPECTRUM computers. When the volume is write-protected, information can be copied only from the volume to computer memory, or to another volume that is write-enabled.

Protection for individual files can also be made by using the RENAME/PROTECT command which protects a file from deletion, replacement, or editing. The file in question appears as a protected file in the directory of the device in which that file resides. The letter P next to the block size number in the directory indicates the file is protected.

3.3 INDIRECT FILES

An indirect file, is a file containing commands that are processed sequentially, yet which could have been entered interactively at a terminal.

The type of operations that RT-11 can best perform in an indirect file are those that involve a fair amount of computer processing without requiring supervision or intervention. For example, multiple assemblies, compilations, and data transfer operations are typical operations for indirect file processing. Also, any series of commands that are likely to be typed often, can easily run as an indirect file.

One of the most common applications for an indirect file is a startup command file which is always called up by the monitor during the bootstrap procedure. The startup command file (either STARTS.COM for single job monitor, STARTF.COM for foreground/background monitor, or STARTX.COM for extended memory monitor) provided by RT-11 contains all the commands required for system startup. These include general device assignments (ASSIGN), possibly one or more SET commands, and usually the DATE command.

3.3.1 Creating an Indirect File

The editor is used to create an indirect file as a text file. The file may be called by any file name but the conventional file type is .COM since this file type is the default used by the monitor to locate the file.

The lines of text that make up the indirect file are structured exactly as if entered as keyboard input. Each command is entered as a single line of text and each command line is terminated with a carriage return (RETURN key).

The following is an example of input that could be used in an indirect file:

DATE	Print the date.
DEASSIGN	Deassign all previous assignments.
ASSIGN TT: LP:	Assign the logical name LP: to the terminal.
ASSIGN DMO: DK:	Assign the Winchester disk (DMO:) as the default device.
SET EDIT TECO	Set the default editor as TECO by invoking TECO.SAV

3.3.2 Executing an Indirect File

As an indirect file can be run under control of the single-job monitor or as the background job of the foreground/background monitor, care must be taken while a foreground job is running that a nondirectory-structured device (eg., magtape) does not have conflicting requests from the two jobs.

The command to start the execution of an indirect file is the @ character followed by the relevant file name (the file type .COM is assumed unless otherwise indicated). Execution starts immediately and the system processes commands in the indirect file in consecutive order. If an error within the indirect file affects the processing of a command, the system prints a system message on the terminal and stops execution of the entire file. Execution of an indirect file may be terminated by typing two CTRL/Cs.

An example for executing an indirect file is as follows:

```
@DEMO.COM
```

3.4 DEVICE HANDLERS

Device handlers are routines that provide the interface to the various hardware devices that are part of the computer system. These handlers drive or service the peripheral devices and control the physical activities of each device. When reference is made to the device by it's physical name, such as LP:, reference is actually being made to it's device handler LP.SYS (Refer Table 3-2 for valid device names).

Each device handler has an allocated register and vector address (refer Spectrum Eleven Hardware Manual), some of which can be changed for different system configurations by using the keyboard monitor SET commands.

All handlers are part of the user's complete system and are tailored for specific versions of an RT-11 operating system. The KMON commands used to manipulate device handlers are as follows:

INSTALL
LOAD
UNLOAD

In addition the REMOVE command removes handlers from the system table. As there is a limited number of free slots available in the system table, it may be necessary to remove handlers that are not being used in order to free up space.

To alter device handler characteristics and certain system configurations, the SET command is used.

3.5 DEVICE STRUCTURES

RT-11 devices are categorized according to two characteristics; the device's method of processing information, and the device's physical structure.

All RT-11 devices are either randomly accessed or sequentially accessed. Random-access devices allow the system to process blocks of data in a random order, ie., independent of the data's physical location on the device or its location relative to any other information. All types of disk fall into this category.

Sequential-access devices require sequential processing of data; the order in which the system processes the data must be the same as the physical order of the data. RT-11 devices that are sequential devices are the card reader, line printer, terminal, null device and magtape.

File-structured devices are those that allow the system to store data under assigned file names. Storage volumes are called file-structured

volumes because they are capable of physically storing files. They can be further categorized as directory-structured and nondirectory-structured volumes based on their method of directory information storage, collection, and printing.

The directory information kept on a volume includes file names and file types, dates of creation, and file lengths. Directory-structured volumes such as disks, diskettes and DECtape keep this information in a single place at the beginning of the volume. Each time a file is added or deleted, the directory information is updated accordingly. A nonstructured-directory volume such as magtape does not keep directory information in any single place on the tape but rather with each individual file. Their directory information is obtained by sequentially reading through all files on the tape and collecting the directory for printing as each file is encountered.

3.6 PHYSICAL DEVICE NAMES

When service from the monitor is requested, a peripheral device must sometimes be specified. Devices are specified by means of a standard two-character physical device name. Table 3-2 lists the most commonly used devices.

RT-11 offers logical as well as physical devices. Two logical devices which have a special meaning to RT-11 are automatically assigned during bootstrap to the device from which bootstrap was initiated.

SY: The device where the system programs (hence SY) reside;
DK: The default file storage device.

These logical devices can both remain assigned to the boot device or DK: can be reassigned to an alternate device. SY: cannot be reassigned and will always be the device booted from (not applicable with TSX-Plus). Normally for dual floppy disk systems, the user's work device is assigned to the second physical device (default file storage device). As an example, during bootstrap which is initiated from SF0: these two logical devices (SY: and DK:) are automatically assigned to SF0: As the user's work device is used in the second drive (SF1:), an assignment must be included in the system startup command file to assign SF1: to DK: The result is that SY: is assigned to SF0: and DK: is assigned to SF1:

When using the keyboard monitor commands, there are several ways of referring to the device.

1. The handler identification may be used, ie., SF for Floppy Disk, DM for Winchester disk, etc. If the unit number is not specified and there is more than one unit for that device, the system assumes unit 0.

2. An assigned device may be used, eg., ASSIGN SF1 DK

3. If no device is stated in the command, the command assumes the default device as initiated during bootstrap.

TABLE 3-2

BA:	Pseudo device for RT-11 BATCH system
CR:*	Card Reader
DD:	TU58 DECtape
DK:	The default logical storage device for all files
DKn:	The specified file storage unit
DM:	RK06/07 Winchester disk
IT:*	Iterator
LP:	Line Printer
LS:	Line Printer (Serial)
MT:*	Magnetic Tape (includes MT4: streaming tape)
MQ:	Message queue pseudo-device for inter job communications under the FB monitor
NL:	Null device
RK:	RK05 Cartridge Disk
SF:*	Spectrum Floppy Disk for all formats including 1D, 2D, 1S, DX, and IBM
SM:*	Spectrum bootable Magtape handler
SY:	The default logical system device which is always the device and unit from which the system is bootstrapped.
SYn:	The specified unit of the same device type as SY:
TT:	Console Terminal
VU:	Virtual Unit

* Special to Spectrum Eleven.

3.7 RT-11 SYSTEM MESSAGES

Message types that can be displayed fall into 5 categories and are in the form:

?Prog-s-Message

where -

Prog is the name of the system program that issued the message

s indicates the severity (category) of the message

Error Severity Levels

I (Information)	Execution continues although the program detected a condition that you should be informed of. The condition may affect execution at a later time and may require future action.
W (Warning)	Execution continues although the program detected a condition that may cause errors in execution. Corrective action may be necessary.
E (Error)	Execution terminates. The program detected an error that will cause other errors during execution. Corrective action is necessary.
F (Fatal/Severe)	Execution terminates. The program did not produce any usable output, and any command or operation depending upon this program output will not execute properly.
U (Unconditional Abort)	Execution terminates. An exetremely serious error occurred that prevents further processing.

The Monitor SET command can be used to change the effect of the severity level on the system's responses. In particular these settings affect the execution of indirect command files (refer SET ERROR NONE).

Reference:

Digital Equipment Corporation
RT-11 System Message Manual
AA-5284C-TC

OVERVIEW

An operating system is a collection of programs and monitors that provide an environment in which programs can be created and run, and which organizes all the hardware and software resources of the computer system into a working unit.

More specifically, the RT-11 Operating System comprises a monitor/executive program for system control and supervision, device handler programs (one for each of the hardware devices), a variety of utility programs for program/data creation and manipulation, and finally the interfaces that are necessary to support several programming language processors.

1.1 MONITOR

The monitor (executive) program is the interface between the system hardware, the system software, and the user. Part of the monitor function is to accept, process and execute user instructions for controlling the system. The monitor program is transferred from the system disk to computer memory during the bootstrapping procedure. A comprehensive set of keyboard monitor commands then allows a user to direct system operations from the console terminal. Chapter 3 describes the monitors, and details the more common keyboard monitor commands.

1.2 UTILITIES AND PROGRAMS

These offer a wide variety of resources and include the handlers that provide the interface between the various hardware that are part of the computer system, and the programs that allow for the creation and editing of text, maintenance of other programs and location of user-programming errors. Refer Table 3-2 (Section 2) for a list of the device handlers, and Section 4 for details of the standard system utilities available with run-time RT-11.

1.3 LANGUAGE PROCESSORS

A language processor is a translating program used to process a source program and one exists for every programming language supported by the system, whether it is a high-level language or a machine-level language. Additional information is available in the following chapter.

2.1 INTRODUCTION

The final important component of the RT-11 operating system is the support that it provides for the various high level language processors. It allows the user to efficiently create a program in any of the high level languages such as FORTRAN, COBOL, BASIC, APL, FOCAL, DIBOL, PASCAL, and many others. The language processor converts program statements into a format suitable for execution. Two of the above mentioned languages, BASIC and APL, are called interpreters. An interpreter translates the instruction into machine code and executes the instruction during the live operation of the routine. On the other hand , the other languages are usually called compilers. This is where the entire program is translated before execution takes place.

High-level languages are considered machine-independent languages because language statements are such that any program written in the language can usually be executed on an entirely different computer system (that supports the language) with few, if any, modifications.

In comparison, machine-level languages, such as the assembly language MACRO-11, require that the user knows how the computer and the peripheral devices work together, as they allow user programs to access and utilize every possible feature available on an RT-11 system. Each computer operation usually requires a separate statement which makes these languages machine-dependent.

The original program, regardless of what language it is written in, is called a Source program. Compilers usually take input from a source program and produce from it another file called the Object file. Depending on the particular language processor being used, the object file will either be directly executable by the computer or will become the source file for a further stage of processing.

The following chart shows the file types allocated to some of the languages and illustrates their differences as language processors:

LANGUAGE	SOURCE	OBJECT
FORTRAN	PROG.FOR Fortran Source Program	PROG.OBJ Object module
COBOL	PROG.COB Cobol Source Program	PROG.SAV Executable code
PASCAL	PROG.PAS Pascal Source Program	PROG.MAC Macro (assembler) code

BASIC	PROG.BAS Basic Source Program	None
MACRO	PROG.MAC Macro code	PROG.OBJ Object module
LINK	PROG.OBJ Object module	PROG.SAV or PROG.REL Executable code

LINK has been included in this table because although it is not a language processor in the strict sense of the word, it is required for some of the others which produce .OBJ output files in order to create executable code.

Reference:

Digital Equipment Corporation
RT-11 Documentation Directory
AA-5285E-TC

INTRODUCTION

Two separate monitors are provided with RT-11 to support both single job (SJ) and foreground/background (FB) modes of processing. Since the FB monitor is actually an extension of the SJ monitor, it provides some commands and programming features that the SJ monitor does not have. These allow the two-job environment to be controlled.

The FB monitor allows two programs to operate concurrently; a foreground program and a background program. Functions such as program development which do not have critical response time requirements, are accomplished in the background which operates when the foreground program cannot run. Both foreground and background are complete RT-11 systems with access to system capabilities. Although they operate independently, foreground and background can communicate through disk files and/or message transmission areas in memory. If a user's application does not require concurrent real-time execution, system resources can be conserved and the SJ monitor used.

The SJ monitor is for users not requiring FB operation or the additional FB features. It supports all hardware devices (except memory management hardware) and runs all system utilities except QUEUE and error logging. If a user's application involves interactive program development, maximum-throughput real-time data acquisition, or continuous execution of a single end-user application program, the SJ monitor should be used. SJ requires less memory and has lower execution overhead. Should the user's requirements change, a properly written program that runs under the SJ monitor can be executed under the FB monitor as a background program with no modifications necessary.

The RT-11 monitors have several powerful features, including English language keyboard commands, device-independent input/output (I/O) programming, a choice of synchronous, asynchronous or event-driven I/O, indirect file command file support, BATCH processing support, and support of up to 8 terminals on the SPECTRUM ELEVEN.

3.1 RT-11 MONITOR COMPONENTS

The RT-11 monitor comprises three major components:

KMON — The Keyboard MONitor controls the interaction between the user and the operating system and is the most visible part of the system software. Among other services, it interprets the monitor command language used to control system operations such as loading and running programs, assigning alternate device names and copying or deleting files. These commands are entered after the monitor prompt.

RMON — Part of the monitor is called the Resident MONitor. It always remains in computer memory regardless of other system operations that may be occurring. Other parts of the monitor are brought into memory from the system volume as needed.

USR — The User Service Routine performs operations related to input and output, such as opening and closing files.

3.2 KMON COMMAND FORMAT

The general syntax for a keyboard monitor command is:

command[/option] input-filespec[/option] output-filespec[/option]

where:

command	is the command name. A command is a keyword that specifies the type of operation to be performed.
/option	represents a command qualifier that specifies the exact action to be taken. Options are used to select some action other than the default and when entered immediately following the command, apply to the entire command string.
input-filespec	represents the device and file name on which the action is to be taken. The input device used is the default file storage device (DK:) unless otherwise specified.
/option	represents a file qualifier that specifies more detailed information about that particular file or action to be taken.
output-filespec	represents the file that is to receive the results of the operation. The output device used is the default file storage device (DK:) unless otherwise specified.
/option	represents a file qualifier that specifies more detailed information about that particular file or action to be taken.

The syntax applicable for a filespec is:

dev:filnam.typ

where:

dev	represents either a logical device name or a physical device name, either of which is a two or three character name as specified in Table 3-2 (Section 2). A colon (:) must be entered after the device name to signify the completion of the name and to separate it from the filenam.
filnam	represents the one to six character alphanumeric name of the file. Each file on a disk (more properly a volume as magnetic tape can contain files as well) has a unique name to identify and distinguish it from all others on a volume.
typ	represents a one to three character alphanumeric file type, some of which are listed in Table 3-1 (Section 2). A file type is used to indicate the file's purpose and must be entered with a preceding period (.) to separate it from the filenam.

3.3 CONVENTIONS AND RULES

The following conventions are used.

1. Capital letters represent command names or options which can be typed in full or in the abbreviated form. The minimum number of characters needed for an abbreviation must be sufficient to make the command or option unique. An easy way to abbreviate the commands, and one that is always correct, is to use the first four characters or if the qualifier starts with NO, the first six characters. In some cases only the first two characters are required.

   Eg., DIRECTORY can be entered as DIR

2. Square brackets enclosing a slash, ie., [/], signify that options are applicable to the command. Only the most common and/or useful options are listed.

   Eg., DIRECTORY/BRIEF (or DIR/BR)

   The square brackets [ ] are not entered and the slash / must be entered.

3. Where several files on the same device are to be specified, they can be enclosed in parentheses.

   Eg., SF0:(TEST1,TEST2,ABC).LST

   The file names are separated by commas.

4. When a command is entered, a carriage return (not shown) signals the monitor to start processing the command.

3.4 WILDCARDS

On occasion, it may be necessary to perform the same utility operation on several files. For example, it may be necessary to copy from one volume to another, all files with a file type of .MAC, or perhaps all file names with the name of TEST may no longer be required. RT-11 provides a short method of performing these functions by using wildcards. The commands accept these wildcards (% and *) in place of the file name, file type, or characters in the file name or file type. The system ignores the contents of the wild field and selects all the files that match the remaining fields.

1. An asterisk (*) can replace a file name:

   Eg., *.MAC

   The system selects all files on device DK: that have a .MAC file type, regardless of their name.

2. An asterisk (*) can replace a file type:

   Eg., TEST.*

   The system selects all files on device DK: that are named TEST regardless of their file type.

3. An asterisk (*) can replace both a file name and a file type:

   Eg., *.*

   The system selects all files on device DK:

4. An embedded asterisk (*) can replace any number of characters in the input file name or file type:

   Eg., A*B.MAC

   The system selects all files on device DK: with a file type of .MAC and where the name starts with an A and ends with a B.

5. The percentage (%) symbol can replace any single character in the input file name or file type:

   Eg., A%B.MAC

   The system selects all files on device DK: with a file type of .MAC and where the file name starts with an A, ends with a B and is three characters long.

Commands supporting wildcards are:

Input file :	Output file :
COPY	COPY
DELETE	RENAME
DIRECTORY	
HELP	
PRINT	
RENAME	
TYPE	

3.5 KEYBOARD MONITOR COMMANDS

This section lists in alphabetic sequence, the more common RT-11 keyboard commands. Each command description includes the command syntax, a table of the most useful valid options (where applicable), some sample command lines, and where relevant pertinent details about the command.

NOTE

Where shown, spaces are mandatory in the command, and in some examples the abbreviated command form is used.

3.5.1 Index of Commands

ASSIGN	Associates a logical device name with a physical device
CLOSE	Makes background output files permanent
COPY	Copies files
CREATE	Creates or extends a file at a specific place
DATE	Sets or displays the current system date
DEASSIGN	Removes logical device name assignments
DELETE	Removes files from a device or queue
DIRECTORY	Lists device or file directories
EDIT	Invokes the text editor
HELP	Lists helpful information
INITIALIZE	Initializes device directories
INSTALL	Adds a new device handler to the system table
LOAD	Makes a device handler permanently resident in memory
PRINT	Prints files on the line printer
R	Loads and runs a memory image file from the System device
REENTER	Starts a program at its reentry address
REMOVE	Removes a device name from the system tables
RENAME	Changes the name of a file
RUN	Loads and runs a memory image file from either DK: or any device stated
SET	Controls various system options
SHOW	Displays system hardware and software status
SQUEEZE	Rearranges disk files to collect unused file space
TIME	Sets or displays the system time
TYPE	Outputs files to the terminal
UNLOAD	Removes a resident device handler from memory

3.5.2 KMON Commands

ASSIGN

The ASSIGN command associates the logical name specified with a physical device.

ASSIGN physical-device-name logical-device-name

The physical-device-name represents the RT-11 standard permanent name that refers to a particular device installed. Refer to Table 3-2 (Section 2).

The logical-device-name represents an alphanumeric name (maximum of three characters) to be assigned to a particular device.

The ASSIGN command simplifies programming. For example an input request could be device IN and the output could be directed to device OUT.

```
ASS SF1 IN
ASS LP OUT
```

where -

SF1 and LP are the physical device names and IN and OUT are the logical device names.

The purpose of the ASSIGN statement is to be able to refer to a device by more than one name.

```
ASS LP LS
```

where either OUT (as in the first example) or LS refer to the physical device LP.

Use the SHOW command to display logical device name assignments on the terminal.

COPY

The COPY command transfers:

one file to another device;
one file to another file;
a number of files to a single file by concatenation;
the contents of a device to another device;
the contents of a bootstrap to a device;
the contents of a device to a file and vice-versa.

COPY[/]
The input-filespecs can be a device name or as many as six input files;

Only one output device or file may be specified.

Commas separate the input files if more than one is specified. However, if the files are to be combined, they are separated by a plus (+) sign.

COPY A.DAT+B.DAT C.DAT

This command combines DK:A.DAT with DK:B.DAT and stores the result in C.DAT

Options :

/BOOT Copies bootstrap information from an RT-11 monitor and device handlers to Block 0 and Block 2-5 of a disk, permitting that disk to then be used as a system disk.

COPY/BOOT SF0:RT11SJ.SYS SF1:

Copy the Bootstrap program RT11SJ.SYS from SF0: to SF1:

/DEVICE Copies block for block, the image of one device to another. No interpretation of files or changing of any information is done. Disks of different sizes can be copied although data will be lost if a large disk is copied to a smaller one. After copying a small disk to a larger one, issue a SQUEEZE command for the larger disk to correctly reset the disk size.

COPY/DEVICE SF0: SF1:

Copy SF0: to SF1:

/EXCLUDE Copies all files on a device except the ones specified.

COPY/EXCLUDE SF0:*.SAV DK:

Copy all files except those with filetype .SAV from SF0: to DK:

/FILES Use with /DEVICE to copy a volume to a file on another volume or vice versa. If magtape is used for the input volume, a file name must be specified. If a file is copied to a volume, the bootstrap and directory of the output volume are replaced by the equivalent blocks of the input file.

COPY/DEVICE/FILES SY: DK:F.DAT

Copy SY: to DK: as file F.DAT

COPY/DEVICE/FILES F.DAT SF0:

Copy file F.DAT to SF0: as a device Volume

/NEWFILES Copies only files with the current date.

COPY/NEWFILES *.* SF1:*.*

The above example shows a convenient way to back up new files after a session.

/POS:n The /POSITION:n option is used when files are copied to or from magtape. As this option applies to the device and not the files, one /POSITION:n option is specified for the output file and one for the input files.

In the following argument definitions, omitting the argument has the same effect as setting n equal to zero. 'n' is interpreted as a decimal number.

Magtape read (copy from tape) operations:

If n is 0:

The tape rewinds and handler searches for specified file. If more than one file is to be copied, a rewind takes place before each search. If the file specification contains a wildcard, the tape rewinds only once and the handler copies all appropriate files.

If n is a positive integer:

The handler looks for the file at file sequence number n and will copy the file it finds there if it is the specified file (or files if using wildcard).

If n is -1:

The handler starts its search from the current position.

Magtape write (copy to tape) operations:

If n is 0:

The tape rewinds before handler copies each file.

If n is a positive integer:

The handler goes to file sequence number n or to logical end of tape, whichever is first. The tape does not rewind between each file copy if more than one file or a wildcard is used.

If n is -1:

The handler goes to logical end of tape, copies the file and does not rewind or check for duplicate file names.

If n is -2:

The tape rewinds between each copy operation and the handler enters the specified file at logical end of tape.

/QUERY Requests file confirmation before a file copy is performed.

COPY/QUERY SY:*.SAV SF1:

This option will request a 'Y' confirmation before it will copy any .SAV file from SY: to SF1: Any other response will not copy that particular file and the copy procedure will continue by setting up the next file's confirmation. This is the default mode.

/NOQUERY This option suppresses the confirmation message that is applicable for some commands. It also suppresses logging of file names if the command contains a wildcard.

COPY/NOQUERY SF0:*.SAV DK:

Copy all .SAV files from SF0: to device DK:

/REPLACE This is the system default mode of operation for the COPY command. If a file exists on the output device with the same name as the file specified for output, the system deletes the duplicate file after the copy operation.

COPY/REPLACE SF0:DEMO.JOB SF1:

Copy DEMO.JOB from SF0: to SF1: and then delete the old version of DEMO.JOB on SF1:

/NOREPLACE Will not copy any file if one already exists on the output device.

COPY/NOREPLACE SF0:DEMO.JOB SF1:

Copy the file DEMO.JOB from SF0: to SF1: only if this file is not on SF1:

/SLOWLY This option transfers files one block at a time. On some devices, a single-block transfer increases the chances of an error-free transfer. Would be used if a previous copy operation failed because of a read or write error.

COPY/SLOWLY SF0:DEMO.JOB SF1:

Copy file DEMO.JOB from SF0: to SF1:

/SYSTEM Allows .SYS files to be copied. If this option is omitted, the .SYS files are excluded from all operations which thus provides a protective feature.

COPY/SYSTEM SY:*.SYS SF1:

Copy all .SYS files from device SY: to SF1:

/WAIT This option is used on systems that have only a single disk drive, or on systems that have dual drives and the system volume is neither the input or output volume. It allows files to be copied from one data disk to another without having to have a system disk installed and waits for the disk to be inserted before commencing the copy operation.

COPY/WAIT DK:DEMO.JOB

Copy file DEMO.JOB from floppy disk to another disk to be inserted in the same drive.

DATE

The DATE command sets or inspects the current system date.

DATE dd-mmm-yy

dd represents the day (a decimal number from 1 to 31);

mmm represents the first three characters of the name of the month;

yy represents the year (a decimal number from 73 to 99).

To enter a date into the system, enter this command as soon as bootstrap is complete. The system uses this date for newly created files, for files transferred to magtape, and for listing files.

```
DATE 15-FEB-82
```

To display the current system date, type the DATE command without an argument:

```
DATE
```

The system date will display on the next line.

DEASSIGN

The DEASSIGN command disassociates a logical device name from a physical device name.

```
DEASSIGN logical-device-name
```

The logical-device-name represents an alphanumeric name, up to three characters in length.

To disassociate all logical names from physical devices, type the DEASSIGN command without an argument.

```
DEASSIGN
```

All logical names are deassigned.

```
DEASSIGN IN:
```

Deassign IN: (from SF1: for example)

DELETE

This command allows files to be deleted. A confirmation is required from the system for each file specified.

```
DELETE[/] filespecs
```

filespecs represents a device with the file or files to be deleted. Up to six files, separated by commas, may be specified.

This command will delete DEMO.JOB and DEMO.BAK from SF1:

```
DEL SF1:DEMO.JOB,DEMO.BAK
```

Options :

/EXCLUDE Deletes all files on a device except the ones specified.

DELETE/EXCLUDE *.BAK

The .BAK files are on the assigned DK device. All files are to be deleted except the .BAK files (and the .SYS files unless the /SYSTEM option is also used).

/NEWFILES Deletes all files with the current date.

DELETE/NEWFILES SF1:*.BAK

This option will delete all .BAK files created with today's date.

/NOQUERY Suppress the confirmation message during the delete function.

DEL/NOQ *.BAK

Deletes all .BAK files on device DK and does not display a confirmation message for each file.

/SYSTEM Allows deletions on .SYS files.

DELETE/SYS DK:*.SYS

Deletes .SYS files from device DK:

/WAIT Deletes a file after allowing a pause for a different disk to be installed.

DELETE/WAIT SF0:DEMO.JOB

This option deletes DEMO.JOB after allowing a pause for the user's work disk to be inserted.

DIRECTORY

The output from a DIRECTORY command can be to a file, a printer, or be displayed on a terminal.

DIR[/] filespecs

filespecs represents the device, file, or group of files for which the directory information is required.

Up to six files may be specified explicitly and the * wildcard character can be used.

As DK: is the default device, when DIR is used with no argument, a directory of DK: is displayed. All directory options will display a summary of the number of files, number of blocks, and the number of free blocks.

Options :

/ALPHA — The ALPHABETIZE option lists the directory of the device specified in alphabetic sequence by file name and file type. It has the same effect as the /ORDER:NAME option.

DIR/ALPHA SF1:

List the directory of SF1: in alphabetic sequence.

/BRIEF — This option lists only file names and file types, omitting file lengths and associated dates.

DIRECTORY/BRIEF

A five column display is produced of the directory on device DK:

/DATE — Will display files with a certain specified date.

DIR/DATE:08:FEB:82 SF0:

All files created on 8th February 1982, on device SF0: will be displayed.

/EXCLUDE — List all files on a directory except those nominated.

DIR/EXCLUDE SY:(*.SAV,*.SYS)

This option lists a directory of all files on the system device except the .SAV and .SYS files.

/FREE — This option is used to print a directory of unused areas and the size of each.

DIR/FREE

Lists all unused areas and the total of free blocks available on device DK:

/FULL — This option lists the entire directory, including unused areas and their sizes in blocks (decimal).

DIR/FULL SF0:

Will display the entire directory for device SF0:

/NEWFILES Lists a directory of files created on the current date.

DIR/NEWFILES DK:

This option lists only the files created on device DK: on the current day.

/ORDER[:category]

This option sorts the directory of the device according to the category specified. The category can be one of the following:

DATE	Chronologically by creation date. Files that have the dame date are sorted alphabetically by file name and type;
NAME	Alphabetically by file name (same as /ALPHABETIZE);
POSITION	Lists files according to their positions on the device (same as using /ORDER with no category);
SIZE	Sequence is based on file size in blocks. Files that are the same size are sorted alphabetically by file name and type;
TYPE	Alphabetically by file type. Files of the same type are sorted alphabetically by file name.

DIR/ORDER:DATE SF1:

As an example for /ORDER, this option will list out the directory for device SF1: in chronological date sequence.

/OUTPUT:filespec

This option is used when specifying a device and filename for a directory output listing file. If the file type is omitted the system uses .DIR

DIR/OUT:SF0:HOLD.DIR

A listing of the directory is output to file HOLD.DIR on device SF0:

/POSITION Lists the file sequence numbers of files stored on magtape.

DIR/POS

/PRINTER This option prints the directory listing on the line printer. The default output device is the terminal.

DIR/FULL/PRINTER

Prints the full directory for device DK: on the line printer.

/REVERSE May be applied to any category listed in the /ORDER option. The display will be in reverse sequence.

DIR/ORDER:DATE SF1:/REVERSE

Will list out the directory for device SF1: in reverse date sequence.

/WAIT Lists a directory of a device after allowing it to be installed.

DIR/WAIT DK:

Lists the directory for device DK: after allowing the disk to be changed.

EDIT

The EDIT command invokes the text editor.

EDIT[/E][/] filespec

E is the text editor. The default editor is EDIT which the system defaults to each time after bootstrap;

filespec represents the file to be edited.

The following command invokes EDIT, opens an existing file, and reads the first page of text.

EDIT DEMO.JOB

The system will invoke the text editor (normally the default editor set by the SET command in the system startup indirect command file).

Options :

/CREATE This option is used to build a new file.

EDIT/CREATE DEMO.TXT

Creates a file DEMO.TXT on device DK:

/EXECUTE:filespec

This option is used with /TECO to execute the TECO commands contained in the file specified with /EXECUTE.

EDIT/TECO/EXECUTE:DEMO.TXT

/INSPECT This option only opens a file for reading and does not create any new output files.

EDIT/INSPECT DEMO.TXT

This command opens the file DEMO.TXT on device DK: for inspection.

/KED This option invokes the Keypad Editor (KED).

EDIT/KED

(For VT100 terminals only).

/K52 This option invokes the Keypad Editor.

EDIT/K52

(For VT52 terminals only)

/OUTPUT:filespec

This option directs text that is edited to the specified file, leaving the input file unchanged.

EDIT/OUTPUT:CHANGE.TXT DEMO.TXT

This command reads file DEMO.TXT and writes the edited text to file CHANGE.TXT .

/TECO This option invokes the TECO editor.

EDIT/TECO

(Can be used on any terminal).

HELP

A HELP file that contains information about the keyboard monitor commands and how to use them, is distributed with RT-11.

HELP[/] topic subtopic:item

topic represents a subject which is a keyboard monitor command;

subtopic represents a category within a topic and can only be SYNTAX, SEMANTICS, OPTIONS or EXAMPLES.

the item represents one member of the subtopic group.

Options : (only two are available):

/PRINTER This option directs output to the line printer.

HELP/PRINTER COPY/QUERY EXAMPLES

Gives an example of the COPY command when used with the QUERY option and directs the output to the printer.

/TERMINAL This is the default operation which lists output to the console terminal.

HELP *

This command lists all the topics in alphabetic sequence.

INITIALIZE

The INITIALIZE command is used to clear and initialize a device directory.

INIT[/] device

device represents the volume requiring initializing.

Use the INITIALIZE command with extreme caution as it destroys any data that may already exist on a device and leaves no files in the directory. Ensure that a backup copy exists of the device. If the command is used with no options, the system initializes the default device directory. If the volume to be initialized has protected files, the system requires confirmation.

INIT SF0:

Initialize the device SF0:

Options :

/BADBLOCKS This option scans a volume for bad blocks and writes .BAD files over them. After the volume is initialized and the scan is completed, the directory consists of only FILE.BAD entries to cover the bad blocks. This procedure ensures that the system will not attempt to access these bad blocks during routine operations. If a bad block is found in either the boot blocks or the volume directory, an error message is printed and the volume is not usable.

INIT/BADBLOCKS DM0:

This command initializes volume DM0: and scans for bad blocks.

/FILE:filespec

This option is used to initialize a magtape and create a bootable tape. For filespec, substitute dev:MBOOT.BOT This file is distributed with RT-11 for this purpose only.

INITIALIZE/FILE:MBOOT.BOT MT0:

/NOQUERY Initializes without displaying a confirmation message.

INIT/NOQ SF1:

Does not require confirmation before proceeding with the initialization of volume SF1:

/RESTORE This option will restore the directory and files that were present on a volume prior to the last initialization, providing no files have been transferred to the volume since the volume was initialized. As RESTORE does not restore the boot blocks, the COPY/BOOT command must be used to make the volume bootable again.

/SEGMENTS:n The number of segments in a disk's directory establishes the number of files that can be created on it. The defaults, with each segment holding 72 files, are:

Disk Type	Segments (Files)	Disk Description
RK	16 (1152)	Cartridge Disk (RK05)
DM	31 (2232)	Winchester Disk (RK07)
DX	1 (72)	DX format Floppy Disk
1S	1 (72)	1S format Floppy DIsk
1D	4 (288)	1D format Floppy Disk
2D	4 (288)	2D format Floppy Disk

INIT/SEG:2 SF1:

Initialize and alter the number of segments for a 2D volume on device DK:

/VOLUMEID[:ONLY]

Writes a volume identification on a device during initialization. The identification consists of a volume ID (up to 12 characters for a block-replaceable device or up to 6 characters for magtape) and an owner name (up to 12 characters for a block-replaceable device or up to 10 characters for magtape).

INIT/VOL SF1:

Initializes and writes a volume identification on device SF1:

The :ONLY option writes a new volume identification without initializing the device. Not valid for magtape.

/WAIT Waits for a different disk to be installed before initializing it.

INIT/WAIT

Initialize the volume to be installed on device DK:

INSTALL

The INSTALL command installs extra handlers into the system table and has to be valid for the version of RT-11.

INSTALL device

device represents the name of the device to be installed.

The INSTALL command accepts no options.

To permanently install a device, the INSTALL command must be included in either the SYSGEN options or in the standard system startup file. This file is invoked automatically when the system is booted. The INSTALL command also allows a configuration for a one-time session without having to reconfigure to revert to the standard device configuration. If there are no device slots (use the SHOW DEVICES command to ascertain this), an existing device must be removed with the REMOVE command.

INSTALL CR:

The card reader is installed in to the system tables from the file CR.SYS (The colon character following the device handler name is optional).

One command can be used to install several devices:

INSTALL CR:,LP:

Install the card reader and the line printer.

LOAD

The LOAD command loads a device handler into memory for use with foreground, background, system jobs, or BATCH. The loading of handlers does use extra memory space but improves system performance on floppy disk based systems.

LOAD device[=jobtype]

device represents the device handler to be made resident;

jobtype assigns the device handler to the background job if it has the value B, or to the foreground or system job if it has the value F. The jobtype specification is invalid with the SJ monitor.

The LOAD command helps control system execution by bringing a device handler into memory and optionally allocating the device to a job. The system then allocates memory for the handler as needed. To change ownership of a device, another LOAD command is used. It is not necessary to first unload the device.

LOAD LP:,MT:=B

The line printer is available for use by either the background, foreground or system job, and magtape belongs to the background job.

LOAD LP:=QUEUE

For a monitor with system job support, the following example reserves the line printer for the system job QUEUE.

Use the SHOW command to display the status of device handlers and device ownership.

PRINT

The PRINT command is virtually the same as the COPY command except that it assumes the line printer (LP:) as the output device.

PRINT[/] filespecs

filespecs represents the file or files to be printed.

Six files may be explicitly specified and separated by commas. The files are printed in the sequence specified in the command line.

Options

/COPIES:n This option prints more than one copy of the file. The argument is a decimal number from 1 to 32 with 1 as the default value. This option must appear immediately after the PRINT command, and not after the file specification.

PRINT/COPIES:3 REPORT

This command prints three copies of file REPORT.LST on the line printer. It is not necessary to specify a file type (extension) as the PRINT command always assumes .LST

/DELETE This option deletes a file after it lists on the line printer. If a confirmation is required use the /QUERY option.

PRINT/DELETE DEMO

Prints DEMO.LST on the line printer then deletes it from device DK:

/NEWFILES Prints all .LST files created on current date.

PRINT/NEW

This option prints all .LST files created on the current date. If a list is required of the .TXT files, it will be necessary to specify the file type: PRINT/NEW *.TXT

/WAIT Waits for alternate disk to be inserted before selecting file for printing.

PRINT/WAIT SF0:DEMO.LST

This command prints DEMO.LST from SF0:

R

The R command loads any file from the system device into memory and attempts to execute it. This command is only for executing .SAV programs from the system disk.

R filespec

filespec represents the program to be executed. The default file type is .SAV

The R command is similar to the RUN command except that the file specified in an R command string must be on the system device (SY:)

R FUTIL

This command loads and executes FUTIL.SAV from device SY:

REMOVE

The REMOVE command removes a device name from the system tables.

REMOVE device

device represents the device to be removed from the system tables.

The REMOVE command accepts no options. Using this command does not change the monitor disk image; it only modifies the system tables of the monitor currently in core. This allows a special system to be configured for a single session without having to reconfigure to return to the standard device configuration. Bootstrapping the system device restores the original device configuration. To permanently remove a device, the REMOVE command must be included in the system startup command file.

The REMOVE command creates a free device slot. A new device can be installed using the INSTALL command. Note that SY: (system device handler), BA: (BATCH handler), and TT:(terminal handler) cannot be removed.

REMOVE LP

Removes the Line Printer handler from the system. It is not mandatory to use a colon.

RENAME

The RENAME command assigns a new name to an existing file.

RENAME[/] input-filespecs output-filespecs

input-filespecs represents the files to be renamed and output-filespecs represents the new name.

Up to six input files may be specified but only one output file. The device specification must be the same for input and output as a file cannot be renamed from one device to another. If a file exists with the same name and file type as the output file specified, the system deletes the existing file unless the /NOREPLACE option is used. RENAME cannot be used for a magtape file (nondirectory-structured device).

```
RENAME *.BAK *.MAC
```

Change all .BAK files to .MAC files on device DK:

Options

/NEWFILES Renames all files having the current date.

```
RENAME/NEWFILES *.BAK *.MAC
```

Rename from .BAK to .MAC the files with today's date.

/PROTECT This option gives a file a protected status so that it cannot be deleted until disabled from the protect status.

```
RENAME/PROTECTION DEMO.JOB
```

File DEMO.JOB will have a protected status.

/NOPROTECT This option enables a file for deletion by disabling the file's protected status.

```
RENAME/NOPROTECT DEMO.JOB
```

DEMO.JOB is no longer a protected file.

/REPLACE Renames file and deletes previous version of specified output file.

```
RENAME/REPLACE SF1:DEMO.JOB DEMO.NEW
```

Rename DEMO.JOB to DEMO.NEW on device SF1: and delete duplicate file DEMO.NEW . REPLACE is the default mode of operation.

/NOREPLACE RENAME/NOREPLACE SF1:DEMO.JOB DEMO.NEW

This option renames DEMO.JOB to DEMO.NEW but if an output file already exists no action occurs and an error message is displayed.

/SETDATE Normally the system preserves the existing file creation date when it renames a file. This option renames files and changes their dates.

RENAME/SETDATE DK:*.BAK *.MAC

Rename all .BAK files to .MAC files on device DK: and set to today's date.

/SYSTEM Renames .SYS files.

RENAME/SYSTEM SF0:AA.SYS BB.SYS

Rename AA.SYS files to BB.SYS files.

/WAIT Waits for different output disk to be inserted before renaming file.

RENAME/WAIT DK:DEMO.TXT NDEM.TXT

Rename DEMO.TXT to NDEM.TXT (on a different disk).

RUN

The RUN command loads a file into memory from the default work device and attempts execution.

RUN filespec [input-list output-list]

filespec represents the program to be executed. The system assumes a .SAV file for the executable file which can reside on any RT-11 block-replacable device. The default device is DK:

The RUN command automatically loads the device handler for the device specified if it is not already resident. It loads a file from a storage device into memory and then begins execution at the program's transfer address.

RUN DEMOPR

This command executes DEMOPR.SAV stored on the default device.

SET

The SET command changes device handler characteristics and certain system configuration parameters.

SET physical-device-name condition

or

SET item condition

physical-device-name represents the device handler with the characteristics that require modification;

item represents a system parameter that needs modification;

the conditions are as specified below.

If more than one condition is applicable for one device, these are separated by commas. With the exception of SET TT, SET USR and the SET item commands, all changes permanently modify the device handlers which remain set even after rebooting. However, if a handler is loaded at the time of the SET command it will be necessary to unload the handler and reload it. For example, UNLOAD SF: LOAD SF: For the SET commands that do not permanently modify the device handlers, the conditions return to the default setting after a reboot. To make these settings permanent, they should be included in the system startup indirect command file.

NB: The colon (:) after each device name is optional.

SET - DEVICE CONDITIONS

The format is SET device/item condition

Examples:
```
SET CR NOHANG
SET EDIT EDIT
SET LP TAB
SET TT CONSOL=1
SET TT NOQUIET
```

Device or Item	Condition	Modification
CARD READER HANDLER		
CR:	CODE=n	Card codes where n is either 26 or 29. The default value is 29.
CR:	CRLF	Appends a carriage return/line feed combination to each card image. This is the normal mode.
CR:	NOCRLF	Transfers each card image without appending a carriage return/line feed combination. The default is CRLF.
CR:	HANG	Waits for a correction if the reader is not ready at the start of a transfer. This is the normal mode.
CR:	NOHANG	Generates an immediate error if the device is not ready at the start of a transfer. The handler waits (regardless of the condition) if the reader is not ready at some point during the transfer, ie., the input hopper is empty but an end-of-file card has not been read. The default is HANG.
CR:	IMAGE	Causes each card column to be stored as a 12-bit binary number, one column per word, and allows the system to read binary card images. The CODE option has no effect in IMAGE mode. Mark-sense cards can be read in this mode and the default is NOIMAGE.
CR:	NOIMAGE	Allows the normal translation (as specified by CODE option) to take place. It translates invalid punch combinations into the error character ASCII backslash (\), which is octal 134. This is the normal mode.
CR:	TRIM	Removes trailing blanks from each card that the system reads. TRIM and NOCRLF should not be used together as card boundaries become difficult to read. This is the normal mode.
CR:	NOTRIM	Transfers a full 80 characters per card. The default is TRIM.

TEXT EDITOR

EDIT	EDIT	Invokes the text editor EDIT with the keyboard monitor EDIT command. This is the normal mode and the system returns to this condition after a reboot.
EDIT	KED	Invokes the keypad editor (KED) with the keyboard EDIT command. Valid only for VT100 terminals. System returns to EDIT EDIT after a reboot.
EDIT	K52	Invokes the keypad editor (K52) with the keyboard EDIT command. Valid only for VT52 terminals. System returns to EDIT EDIT after a reboot.
EDIT	TECO	Invokes the text editor TECO with the keyboard EDIT command. System returns to EDIT EDIT after a reboot.

ERROR MESSAGES

ERROR	WARNING	Causes indirect command files and keyboard monitor commands to abort if warnings, errors, or severe errors occur. Warning errors contain the -W- characters and the system defaults to ERROR ERROR after a reboot.
ERROR	ERROR	Causes indirect command files and certain keyboard monitor commands to abort if errors or severe or fatal errors occur. Error messages contain the -E- characters. This is the normal setting and the system returns to this condition after a reboot.
ERROR	SEVERE	Causes indirect command files and keyboard monitor commands to abort only if severe or fatal errors occur. Severe error messages contain the -F- characters and the system defaults to ERROR ERROR after a reboot.
ERROR	NONE	Allows indirect command files and keyboard monitor commands to continue to execute even though they contain significant errors. Fatal error messages contain the -U- characters and these will still cause the program to abort. The system defaults to ERROR ERROR after a reboot.

LINE PRINTER HANDLER

LP: and LS: both use the same conditions in the SET commands. If LP has been assigned as LS, then the SET commands use LP:. Consequently, if LS has been assigned as LP, the SET commands use LS:

LP:	CR	Sends carriage returns to the printer. To allow overstriking on the printer, use this condition for any FORTRAN program that uses formatted input and output. System defaults to NOCR.
LP:	NOCR	Prevents the system from sending carriage returns to the printer. This is the default setting.
LP:	CSR=n	Modifies the line printer handler to use n as the contents of the CSR (Control and Status register). Value must be an octal address not less than 160000.
LP:	CTRL	Passes all characters, including nonprinting control characters (tabs, bells, line feeds, etc.) to the printer. The default is NOCTRL.
LP:	NOCTRL	Ignores nonprinting control characters. This is the default mode.
LP:	FORM	Causes the line printer handler to generate hardware form feeds. This is the default setting.
LP:	NOFORM	Causes the line printer to use consecutive line feeds at page perforations to simulate a hardware form feed. LENGTH=n setting must also be used and the paper positioned at top of form. The default is FORM.
LP:	FORM0	Issues a form feed before a request to print block 0. This is the default mode.
LP:	NOFORM0	Turns off FORM0 mode.
LP:	HANG	Waits for a correction to be made if the line printer is not ready, either before or during printing. This is the default mode.

LP:	NOHANG	Generates an immediate error if the line printer is not ready. The default is HANG.
LP:	LC	Allows the system to send lower-case characters to the printer.
LP:	NOLC	Translates lower-case characters to upper case before printing.
LP:	LENGTH=n	Causes the line printer to use n as the number of lines per page. This option is used with the NOFORM and SKIP=n settings. The default number of lines is 66.
LP:	SKIP=n	Causes the line printer handler to send a form feed to the printer when it comes within n lines of the bottom of the page. Use with NOFORM and LENGTH=n settings, and to disable a condition, set SKIP=0.
LP:	TAB	Sends TAB characters to the line printer. The default is NOTAB.
LP:	NOTAB	Does not send TAB characters to the printer; simulates TABs with spaces.
LP:	VECTOR=n	Modifies the line printer handler to use n as the vector of the line printer controller. The value for n must be an even octal address below 500. Saves modifying the handler source code.
LP:	WIDTH=n	Sets the line printer width to n, where n is a decimal integer between 30 and 255 inclusive. The default is 132.

MAGTAPE HANDLER

MT:	ODDPAR	Sets parity to odd.
MT:	NOODDPAR	Sets parity to even.

CONSOLE TERMINAL HANDLER

TT:	CONSOL=n	Changes background console to the terminal associated with logical unit number n. Valid only for multi-terminal configuration and invalid for a remote line. The system returns to this default after a reboot.
TT:	CRLF	Issues a carriage return/line feed combination on the console terminal when past the right margin. Setting is invalid with a non-multi-terminal SJ monitor. This is the default mode after a reboot.
TT:	NOCRLF	Takes no special action at the right margin. Setting is invalid with a non-multi-terminal SJ monitor. The default is CRLF.
TT:	FB	Treats CTRL/B and CTRL/F (and CTRL/X in system job monitors) as background and foreground program control characters which are not transmitted to programs. Not valid for SJ monitor. This is the default mode.
TT:	NOFB	Causes CTRL/B and CTRL/F (and CTRL/X in system job monitors) to have no special meaning. Not valid for SJ monitor. The default mode is FB.
TT:	FORM	Indicates that the console terminal is capable of executing hardware form feeds. This setting is invalid with a non-multi-terminal SJ monitor. The default is NOFORM.
TT:	NOFORM	Simulates form feeds by generating eight line feeds. This setting is not valid for a non-multi-terminal SJ monitor. This is the default mode.
TT:	HOLD	Hold screen mode for VT50/52/61 terminals. The default setting is NOHOLD.
TT:	NOHOLD	Disables the Hold screen mode for VT50/52/61 terminals. This is the default setting.
TT:	PAGE	Treats CTRL/S and CTRL/Q characters as terminal output hold and unhold flags which are not transmitted to programs. Must be used with VT100 terminals. This setting is not valid for the non-multi-terminal SJ monitor. This is the default setting.

TT:	NOPAGE	Causes CTRL/S and CTRL/Q to have no special meaning. This setting is not valid for the non-multi-terminal SJ monitor. The default is PAGE.
TT:	QUIET	Prevents the system from echoing lines from indirect files. The default is NOQUIET.
TT:	NOQUIET	Echoes lines from indirect files. This is the default setting.
TT:	SCOPE	Echoes RUBOUT/DELETE characters as backspace-space-backspace. The default is NOSCOPE.
TT:	NOSCOPE	Echoes RUBOUT/DELETE characters by enclosing the deleted characters in backslashes. This is the default mode.
TT:	TAB	Indicates that the console terminal is capable of executing hardware tabs. This setting is not valid for the non-multi-terminal SJ monitor. The default is NOTAB.
TT:	NOTAB	Simulates tab stops every eight positions. This setting is not valid for the non-multi-terminal SJ monitor. This is the default mode.
TT:	WIDTH=n	Sets the terminal width to n, where n is an integer between 30 and 255. This setting is not valid for the non-multi-terminal SJ monitor. The system initially sets the width to 80 which is the default value after a reboot.

USER SERVICE ROUTINE

USR	SWAP	Allows the background job to place the USR (user service routine) in a swapping state. This is the normal mode for FB and SJ monitors and is not valid for an XM monitor.
USR	NOSWAP	Prevents the background job from placing the USR in a swapping state. This setting is not valid for an XM monitor and the default is SWAP for FB and SJ monitors.

WILDCARDS

WILD	EXPLICIT	Causes the system to recognize file specifications exactly as they are typed. If a file name or a file type is omitted in a file specification, the missing item will not be replaced with an asterisk. The default is IMPLICIT.
WILD	IMPLICIT	Causes the system to interpret missing fields in file specifications as asterisks (*). This is the default mode.

SHOW

The SHOW command displays details of the system.

SHOW X

where X can be a hardware configuration, monitor version, special features in effect, device names and logical device name assignments, terminal characteristics for terminals currently active on a multi-terminal system, or device handler status.

If SHOW is specified without an option, the system's device assignments are displayed. These devices are those known by the RT-11 monitor currently running in memory and any additions made with the INSTALL and REMOVE commands is reflected.

A SHOW command could provide information as shown in the following example:

```
TT
SF        (Resident)
          SF0 = SY , DK
          SF1 = IN , DK1
LP   =  LS , OUT
LS        (Loaded)
BA
NL
6 free slots
.
```

This shows that SF is resident in memory. The other device handlers known to the system are: TT, LP, LS, BA and NL. There are six free slots in the table. SF0: has the logical names SY and DK. SF1: has the logical names IN and DK1. LP has the logical names LS and OUT. The LS handler is loaded.

Options

Note that these options are preceded by a space after the command SHOW. A backslash (/) is not applicable.

ALL This option is a combination of CONFIGURATION, DEVICES, JOBS, and TERMINALS, in that order. The ALL option also tests the device assignments.

CONF The CONFIGURATION option displays the monitor version number and patch level, the monitor SET options in effect, the hardware configuration, and the special features in effect (if any).

A typical configuration could be as follows:

```
RT11SJ (S) V04.00

Booted from SF0:RT11S2
Resident Monitor base is 151144 (53860.)
USR is set SWAP
TT is set NOQUIET
Indirect File abort level is ERROR
Indirect File nesting depth is 3

LSI-11 Processor
Extended Instruction Set (EIS)
Floating Instruction Set (FIS)
50 cycle System Clock

Device I/O time out support
Multi-terminal support
SJ time support
```

First the system shows the version number and patch level of the currently running monitor.

Next, information about the system is displayed. The first line is the device from which the system was bootstrapped. The next line prints the resident monitor's base address, in octal. Then it shows whether the user service routine (USR) is SWAP or NOSWAP and if TT is QUIET or NOQUIET. The Indirect File abort level could be set to NONE, WARNING, ERROR or SEVERE and the nesting level displays as a decimal number.

The next lines of information include the processor type and other items present on the system including instruction sets and the clock frequency.

Finally, any special features are shown.

DEVICES This option displays the RT-11 device handlers, their status and their vectors. The status can be either Installed, Not installed or Resident.

JOBS This option displays data about the jobs that are currently loaded by detailing:

The job name and number;

The console the job owns (only if multi-terminal monitor);

The priority level of the job;

The job's running state;

The low and high memory limits of the job;

the start address of the job's impure area.

TERMINALS The option indicates the status of and special features in effect for currently active terminals on multi-terminal systems only.

SQUEEZE

The SQUEEZE command consolidates all unused blocks and directory entries.

SQUEEZE[/] device

device represents the block replaceable volume to be compressed.

To perform a squeeze operation, the system moves all the files to the beginning of the specified device, producing a single unused area after the group of files. The Squeeze operation does not change the bootstrap blocks of a device or move files with .BAD file types.

The SQUEEZE command should be used on a 'directory full' error even if there is space remaining on the volume.

SQUEEZE SF1:

This would consolidate the space on device SF1:

/OUTPUT:filespec

This option is used to transfer all the files from the input device to the output device in compressed format, an operation that leaves the input device unchanged. The output device must be an initialized block replaceable volume (INIT command).

SQUEEZE/OUTPUT:RK1: RK0:

This option transfers all the files from RK0: to RK1: in compressed format, leaving RK0: unchanged.

/NOQUERY Squeezes the files on disk without displaying confirmation messages.

SQU/NOQ SF1:

This option suppresses the confirmation message and compresses all the files on SF1:

/WAIT Waits for alternate disk to be installed before commencing the squeeze operation.

SQUEEZE/WAIT RK0:

This compresses the files on device RK0: The system may repeat the 'Mount input volume - Mount output volume' cycle several times to complete the SQUEEZE operation.

TIME

The TIME command enables the time to be set or displayed.

TIME hh:mm:ss

hh represents hours (from 0 to 23);
mm represents minutes (from 0 to 59);
ss represents seconds (from 0 to 59).

The system keeps time on a 24-hour clock.

To enter the time of day, the time is specified in the above format. This should be entered as soon as bootstrap is complete.

To enter the time of day:

TIME 13:15:00 (1:15 pm)

To display the current time of day, use the TIME command without an argument. The actual time will only display if the CLOCK switch is on and the correct time was entered after bootstrap.

TYPE

This is similar to the PRINT command but outputs the contents of a file to the terminal.

TYPE[/] filespec

filespec represents the file or files to be typed.

Six files, separated by commas, may be specified as input to the TYPE command and these are displayed in the sequence entered. If the file type is omitted, the system assumes .LST.

TYPE DEMO

The contents of the file DEMO.LST on device DK are displayed on the terminal screen.

Options :

/DELETE This option will delete a file after it has typed on the terminal.

TYPE/DELETE SF1:DEMO.LST

Program DEMO.LST is deleted from SF1: after it has displayed on the terminal.

/NEWFILES Types on the screen, all files created with the current date.

TYPE/NEWFILES *.LST

Display all new files after a day's session on the computer.

/WAIT Waits for alternate disk to be installed before selecting file for typeout.

TYPE/WAIT SF1:DEMO.LST

DEMO.JOB types to the terminal from the disk installed in SF1:

UNLOAD

The UNLOAD command removes handlers or jobs which are no longer required by the user. Handlers or jobs which are active cannot be unloaded.

UNLOAD device

device represents the device handler to be unloaded.

UNLOAD clears ownership for all units of the device type specified but the handler itself remains resident. On receipt of an UNLOAD command, the system returns any memory it frees to a free memory list which is reclaimed by the background job.

UNLOAD F

This command unloads the foreground job and frees the memory it occupied.

UNLOAD RK:

This command clears ownership of all units of RK: If RK: is the system device, the RK handler itself remains resident.

Reference:

Digital Equipment Corporation
RT-11 System User's Guide
AA-5279B-TC

INTRODUCTION

The Command String Interpreter (CSI) is a utility interpreter which is part of the User Service Routine. It is used by many of the system utility programs (refer next chapter) and by some of the Spectrum RT-11 utilities (eg., SAVRES and WUTIL). It accepts a line of ASCII input and interprets it as a string of input specifications, output specifications, and options for use by these utility programs.

In response to the system prompt, the user enters a Run command and the name of the program to be run. The response is a command level prompt for file specifications:

```
.R PIP     run the Peripheral Interchange Program
*          the CSI is now ready to accept a specification string.
```

1.1 SYNTAX

The specifications are in the following general syntax:

```
output-filespecs/option=input-filespecs/option
```

The syntax for output-filespecs is:

```
dev:filnam.typ[n],..dev:filnam.typ[n]
```

The syntax for input-filespecs is:

```
dev:filnam.typ,..dev.filname.typ
```

The syntax for /option is:

```
/o[:oval] or /o[:dval.]
```

where -

dev: represents either a logical device name or a physical device name, which is a two- or three- character name from Table 3-2 (Section 2).

If a device name is not supplied, the system uses device DK: The device specified for the first file in a list of input or output files, applies to all files in that list until a different device name is supplied. For example:

*DMO:DEMO.JOB,LP:=TEST.1,SF1:TEST.2,TEST.3

This command is interpreted as follows:

*DMO:DEMO.JOB,LP:=DK:TEST.1,SF1:TEST.2,SF1:TEST.3

File DEMO.JOB is stored on device DMO:
File TEST.1 is stored on default device DK:
Files TEST.2 and TEST.3 are stored on device SF1:

filnam.typ — is the name of a file (from one to six alphanumeric characters followed optionally by a period and up to a three character file type). If the file type is omitted, the system may apply a default file type that the program specifies. Tabs and spaces are not permitted and up to three output and six input files are allowed.

[n] — is optional and stands for the number of blocks required for an output file. n is a decimal number (up to 65,535) enclosed in square brackets and immediately follows the output filnam.typ to which it applies.

/o[:oval] or /o[:dval.]

is one or more options. These are not normally required in an I/O command string, since most programs assume default values for any option. The functions of these options vary according to the program being used:

oval is either an octal number or one to three alphanumeric characters (the first of which must be alphabetic) that the program converts to Radix-50 characters.

dval is a decimal number followed by a decimal point.

A minus (-) sign can be used to denote negative octal or decimal numbers.

Generally, these options and their associated values (if any), should follow the device and file name to which they apply. If the same option is to be repeated several times, but with different values, the option is only entered once and each value is separated by a colon. Octal, Radix-50 and decimal values may be mixed.

= is a delimiter that separates the output and input fields. This separator may be omitted if there are no output files. The < sign may be used in place of the = sign.

To return to the monitor level, the user must type CTRL/C.

Reference:

Digital Equipment Corporation
RT-11 System User's Guide
AA-5279B-TC

OVERVIEW

There are many utilities and programs supplied with run-time RT-11. These include text editors, system utility programs, and Webster supported utilities for various types of disks and magtape.

Text editors allow the user to create and modify any type of text. Instructions for two of these editors (EDIT and TECO) are included in this section.

System utility programs allow for the manipulation and maintenance of programs and data; for transfers between devices, and for deletions when required. The user may invoke these utility programs (PIP, DUP, DIR, FILEX and RESORC) directly instead of indirectly through the keyboard monitor commands.

2.1 SYSTEM UTILITIES

This section briefly outlines the file management utility programs available with run-time RT-11. Nearly all the capabilities of RT-11 can be accessed through the keyboard monitor commands (refer Section 3) but it is the utility programs that offer a wider range of facilities and actually perform many of the system's functions. For example, When the COPY command is issued, the utility programs PIP, DUP or FILEX perform the various options of the copy procedure.

This method of obtaining services from RT-11 is more complex than the keyboard monitor commands as it does not use the English language command style and the syntax is quite different. Chapter 1, 'Command String Interpreter', describes the general syntax of the input/output specification string that the utility programs accept.

The five system utility programs offered with run-time RT-11 are:

PIP	Peripheral Interchange Program. Used to transfer files between any valid RT-11 devices, and to merge, rename and delete files.
DUP	Device Utility Program. Used for any file and disk maintenance.
DIR	Directory. Used for directory listing operations.
FILEX	File Exchange Program. Converts files from one format to another, to be used with various operating systems.
RESORC	Resource Utility Program. Displays system information on the terminal.

2.1.1 Peripheral Interchange Program (PIP.SAV)

PIP is a file transfer and file maintenance utility program which can be used to transfer files between any valid RT-11 device, and to merge, rename and delete files. PIP also allows files to be protected against accidental deletion. File protection is indicated by the letter 'P' next to the file size as listed in the file's directory.

The keyboard monitor commands that call in this program are:

COPY
DELETE
PRINT
RENAME
TYPE

2.1.2 Device Utility Program (DUP.SAV)

DUP performs general device tasks such as initializing devices, scanning for bad blocks, duplicating device contents, and reorganizing files on the device. It operates only on RT-11 file structured devices.

The keyboard monitor commands that call in this program are:

BOOT
COPY
CREATE
DIRECTORY
INITIALIZE
SQUEEZE

2.1.3 The Directory Program (DIR.SAV)

The Directory Program performs a wide range of directory listing operations. It can list directory information about a specific device in summarized form or in detailed form where the file name, file type, creation date, and block size is given. DIR can also organize its listings in several ways, such as alphabetically or chronologically.

The keyboard monitor command used to call in the DIR program is:

DIRECTORY

2.1.4 File Exchange Program (FILEX.SAV)

FILEX is a general transfer program that converts files from one format to another between various operating systems. Transfers can be initiated between any block-replaceable RT-11 directory-structured device and any of the following supported devices:

PDP-11 DOS/BATCH DECtape

* DOS/BATCH disk
RSTS DECtape
* DECsystem-10 DECtape
Interchange diskette (RX01, RX02 single-density)

* Denotes valid as input only; others valid as input and output.

FILEX does not support magtape or double-density floppy disk and only one file at a time can be transferred to interchange floppy disk format. Three data formats may be used: ASCII, image, and packed image. In ASCII mode, FILEX deletes null and rubout characters as well as parity bits.

The keyboard monitor commands used to call in FILEX are:

COPY
DELETE
DIRECTORY
INITIALIZE

2.1.5 Resource Utility Program (RESORC.SAV)

RESORC can display the following data about a system.

Monitor version number;
SET options in effect;
Hardware configuration;
Currently loaded jobs;
System generation special features in effect;
Device aassignments;
Device handler status;
Status of currently active terminals on multi-terminal systems;

The keyboard monitor command used to call in RESORC is:

SHOW

INTRODUCTION

TECO is a powerful text editing program that runs under RT-11. TECO is an editor which only makes sequential edits to a file, ie., a pipeline editor. The file to be edited is read into the text buffer one portion at a time, is edited and then written out. Once a portion of text has been written out, further editing is not possible unless the output file is reopened in a later edit as a new file to be edited.

TECO may be used to edit any form of ASCII text, program sources, manuscripts or correspondence. Since TECO is a character oriented editor rather than a line editor, text edited with TECO does not have line numbers associated with it and it is not necessary to replace an entire line of text in order to change one character.

TECO considers any string of ASCII codes to be text. Text is broken down into units of characters, lines, and pages. A character is one ASCII code. A line of text is a string of ASCII codes including one line terminator (usually a line feed) as the last character on the line. A page of text is a string of ASCII codes including one form feed character as the last character on the page.

TECO maintains a text buffer in which text is stored. The buffer usually contains one page of text, but the terminating form feed character never appears in the buffer. TECO also maintains a text buffer pointer. The pointer is a movable position indicator which is never located directly on a character, but is always between characters, ie., between two characters in the buffer, before the first character in the buffer, or after the last character in the buffer.

Line feed and form feed characters are inserted automatically by TECO. A line feed is automatically appended to every carriage-return typed to TECO and a form feed is appended to the contents of the buffer by certain output commands. Additional line feed and form feed characters may be entered into the buffer as text. If a form feed character is entered into the buffer, it will cause a page break upon output and text following the form feed will begin a new page.

Finally, TECO maintains an input file and an output file, both of which are selected by the user through file specification commands. The input file may be on any device from which text may be accepted. Text is written from the input file into the text buffer, and is written from the buffer onto the output file. The output file may be on any device on which edited text may be written. Once text has been written to the output file, it cannot be accessed again without closing the output file and reopening it as an input file.

3.1 TECO COMMANDS

TECO may be called from command level by typing R TECO (for RT-11), folowed by a carriage-return. TECO will respond by printing an asterisk at the left margin to indicate that it is ready to accept user commands.

A TECO command consists of one or two characters, optionally preceded by a numeric argument and sometimes followed by a text argument. In addition there are several immediate action commands which are discussed later on in this chapter. TECO commands may be strung together by concatenation and although no delimiter is necessary, a single ESCAPE may be inserted between the commands. This ESCAPE will keep numeric values generated by the preceding command from affecting the commands that follow the ESCAPE. TECO commands are accumulated into a command string as they are typed and this string is executed when it is terminated by typing two consecutive ESCAPE characters. TECO then executes the commands in the order in which they appear in the command string, until the string is exhausted or an error occurs.

3.2 TECO COMMAND STRUCTURE

The general syntax for TECO commands is -

[n]C[text]$$

where -

n — represents a numeric argument. Some numeric arguments must be positive; others can be negative or zero. A numeric argument could specify -

the number of characters that a command is to act on;

the number of times the comnmand is to be performed;

the number of lines to be effected.

C — represents a one or two letter command (refer to the Command Summary).

text — represents a string of ASCII characters terminated by two ESCAPE characters.

3.3 TECO COMMAND SUMMARY

TECO commands fall into nine general categories. Each of these categories is discussed separately and where applicable, sample commands are given.

Input file specification
Output file specification
Closing files
Input and Output commands
Pointer positioning commands
Type out commands
Text Deletion commands
Text Insertion command
Search commands

Input File Specification

TECO will accept input text from any input device in the operating system. The input device may be specified by means of an ER command terminated by an ESCAPE. The ER command causes TECO to open the specified file and print an error message if the file is not found. This command does not cause any portion of the file to be read into the text buffer.

Teco will only keep one input and one output file open and selected at a time. The current input file may be changed by simply using the ER command to specify a new file.

It is not always necessary to specify an input file. Commands may be typed to insert the text directly into the text buffer and at the end of each page, the contents of the buffer can be written onto an output file. As all input is supplied from the keyboard, no input file is necessary.

ERfilespec$ Edit Read. Opens a file for input.

ERSF1:TEST.JOB$

Prepare to read input file TEST.JOB from SF1:

Output File Specification

TECO will write output text onto any output device in the operating system. The output file may be specified by means of an EW command

terminated by an ESCAPE. If the output file is a file-structured device, the file name and any extension must be supplied. If a filename is specified but no device is defined, the system's default device is assumed.

When the contents of an input file are to be reviewed, ie., no editing is to take place, it is not necessary to specify an output file. The contents of the input file are read into the text buffer, page by page, and examined at the terminal. As all output is printed on the user terminal, no output file is needed.

EWfilespec$ — Edit Write. Opens a file for output.

EWDM0:TEST.LST

Prepare to write output file TEST.LST on DM0:

EBfilespec$ — Edit Backup. This command is recommended for most editing jobs. It is used for files on file-structured devices only. It opens the specified file for input and the specified file for output. This command also keeps the latest copy of the input file available to the user.

ERSF1:INPUT.JOB$EWOUTPUT.JOB$$

Opens an input file INPUT.JOB which is on SF1: and opens an output file OUTPUT.JOB. The double ESCAPE ($$) terminates the command string and causes the string to be executed. The ESCAPE which terminates the EW command can be one of the ESCAPEs which terminates the command string.

Closing Files

After editing is complete, the EX command is used to close out a file and exit from TECO. The current contents of the text buffer and any portion of the input file that has not yet been read are copied to the output file before TECO exits. The EX command takes no arguments.

EX — Writes the text buffer to the current output file, moves the remainder of the current input file to the current output file, closes the output file, then returns to the operating system.

```
ERINPUT.JOB$EWOUTPUT.JOB$EX$$
```

Opens an input file INPUT.JOB and opens an output file OUTPUT.JOB, then copies all the text in the input file to the output file, closes the output file, and exits from TECO.

Input and Output Commands

The following commands permit pages of text to be written into the text buffer from an input file or written from the buffer onto the output file. Once a page of text has been written onto the output file, it cannot be recalled into the text buffer unless the output file is closed and reopened as an input file. After each of the above commands, TECO positions the pointer before the first character in the buffer.

A	Appends the next page of the input file to the contents of the text buffer. This combines the two pages of text on a single page with no intervening form feed character.
Y	Clears the text buffer, then reads the next page of the input file into the buffer. Since the Y command causes the contents of the text buffer to be lost, it is not permitted if an output file is open and there is text in the buffer.
P	Writes the contents of the text buffer onto the next page of the output file, then clears the buffer and reads the next page of the input file into the buffer.
nP	Executes the P command n times, where n must be a positive (non-zero) integer. If n is not specified, a value of 1 is assumed.

Pointer Positioning Commands

The buffer pointer provides the means of specifying the location within a block of text at which insertions, deletions or corrections are to be made. The following commands permit the buffer pointer to be moved to a position between two adjacent characters in the buffer.

These commands may be used to move the buffer pointer across any number of lines or characters in either direction; however, they will not move the pointer across a page boundary. If a C command attempts

to move the pointer backward beyond the beginning of the buffer or forward past the end of the buffer, an error message is printed and the command is ignored. If an L command attempts to exceed the page boundaries in this manner, the pointer is positioned at the boundary which would have been exceeded and no error message is printed.

J	Moves the pointer to a position immediately preceding the first character in the buffer.
nJ	Moves the pointer to a position immediately following the nth character in the buffer.
ZJ	Moves the pointer to a position immediately following the last character in the buffer.
L	Moves the pointer forward across the next line terminator (line feed, vertical tab or form feed) and positions it at the beginning of the next line.
nL	Executes the L command n times. A positive value of n advances the pointer to the beginning of the nth line following its current position. A negative value of n moves the pointer backwards to the beginning of the nth complete line preceding its current position. If n is zero, ie., the command is 0L, the pointer is moved to the beginning of the line on which it is currently positioned.
C	Advances the pointer forward across one character.
nC	Executes the C command n times. If n is positive, the pointer is moved forward across n characters. If n is negative, the pointer is moved backward across n characters. If n is zero, the pointer position is not changed. Note that there are two characters, <CR> and <LF>, at the end of each line in the buffer.

Type Out Commands

These commands are used to type out part of or all of the contents of the buffer for examination. These commands do not move the buffer pointer.

T	Types out the contents of the text buffer from the current position of the pointer through and including the next line feed character.
nT	Types n lines. If n is positive, types n lines following the current position of the pointer. If n is negative, types n lines preceding the pointer. If n is zero, ie., the command is 0T, the contents of the buffer are typed from the beginning of the line on which the pointer is located, up to the pointer. This is useful for verifying the location of the buffer pointer.
HT	Types the entire contents of the text buffer.
V	Types out the current line.

There are three special control commands which can be used while TECO is typing out text to the terminal. If these commands are used in a command string, they do not have the same effect as described in this paragraph.

<CTRL/O>	Stops the terminal output of the current command string. TECO continues to run and to send characters to the terminal but they are suppressed from actually printing on the terminal. The characters can resume printing to the terminal by typing another <CTRL/O> while type out is being supressed.
<CTRL/S>	This command freezes the terminal output of the current command string. TECO stops running the next time it tries to output a character to the terminal. A <CTRL/Q> command will allow the printout to be resumed.
<CTRL/Q>	Causes TECO to resume any type out that was frozen by the <CTRL/S> command. This command will only have an effect while the type out is frozen.

Text Deletion Commands

The following commands permit deletion of single characters, groups of adjacant characters, single lines, or groups of adjacent lines. Like the L and C commands, the K and D commands may not execute across page boundaries. If a K command attempts to delete text up to

and across the beginning or end of the buffer, text will be deleted only up to the buffer boundary and the pointer will be positioned at the boundary. No error message is printed. A D command attempting to delete text past the end or beginning of the text buffer will produce an error message and the command will be ignored.

D	Deletes the first character following the current position of the buffer pointer.
nD	Executes the D command n times. If n is positive, the n characters following the current pointer position are deleted. If n is negative, the n characters preceding the current pointer position are deleted. If n is zero,the command is ignored.
K	Deletes the contents of the text buffer from the current position of the pointer up to and including the next line feed character.
nK	Executes the K command n times. If n is positive, the n lines following the current pointer position are deleted. If n is negative, the n lines preceding the current pointer pointer position are deleted. If n is zero, the contents of the buffer from the beginning of the line on which the pointer is located up to the pointer are deleted. It is not an error if more lines are specified than occur when a boundary of the text buffer is encountered.
HK	Deletes the entire contents of the text buffer.

Text Insertion Commands

This command enables the string of characters specified in the command to be inserted into the text buffer at the current position of the buffer pointer. Following the execution of the insert command, the pointer will be positioned immediately after the last character of the insertion.

The length of an insertion command is limited primarily by the amount of memory available for command string storage. During normal editing, it is advisable to limit insertions to approximately 15 lines each. When command string space is due to run out, TECO will ring the terminal's bell and then allow the typing of a further 10 characters. It may be necessary to shorten a command string by use of the DELETE key.

Itext$ Where 'text' is a string of ASCII characters terminated by an ESCAPE. The specified text is inserted into the buffer at the current position of the pointer. The pointer is positioned immediately after the last character of the insertion.

Search Commands

Search commands cause TECO to scan through text until a specified string of characters is found, and then position the buffer pointer at the end of the string. A character string search begins at the current position of the pointer. It proceeds within the current buffer in a forward or a reverse direction, or through the file in a forward direction. Specifying a negative numeric argument to the search command causes the search to proceed backwards from the pointer.

The last explicitly specified search string is always remembered by TECO. Therefore if a search command is specified with a null search string argument, the last explicitly defined search string will be used. This saves the necessity of retyping a complex or lengthy search string on successive search commands.

Normally searches are 'unbounded'. That is they search from the current position to the end of the text buffer (or if a backward search, the beginning of the buffer). A bounded search will only search from the current position to the specified bound limit. If the search string is found within the bound limits, the pointer is positioned immediately after the last character in the string. If the string cannot be found, the pointer is left unchanged.

Stext$ Where 'text' is a string of ASCII characters terminated by an ESCAPE. This command searches the text buffer for the next occurrence of the specified character string following the current pointer position. If the string is found, the pointer is positioned after the last character on the string. If it is not found, the pointer is positioned immediately before the first character in the buffer and an error message is printed.

nStext$ This command searches for the nth occurrence of the specified character string, where n is greater than zero.

-nStext$ Identical to 'nStext$' except that the search proceeds in the reverse direction. If the string is not found, the pointer is positioned

immediately before the first character in the buffer and an error message is printed.

Ntext$ — Performs the same function as the S command except that the search is continued across page boundaries, if necessary, until the character string is found or the end of the input file is reached. If the end of the input file is reached, an error message is printed. It will then be necessary to close the output file and re-open it as an input file before any further editing may be done on that file.

nNtext$ — This command searches for the nth occurrence of the specified character string, where n must be greater than zero.

There are also several Search with Replace commands. These commands perform equivalent functions to the search commands but replace the searched text with other text.

FStext1$text2$ — Searches for text1 and if found will replace it with text2.

FNtext1$text2$ — Same as previous example except that the search is continued across page boundaries if necessary.

3.4 IMMEDIATE ACTION CONTROL COMMANDS

There are certain immediate action control commands available for TECO that take immediate effect and are used to edit a command string as it is being entered. A control command is activated by holding down the CTRL key with the character key. The echo notation for each control/character is specified to enable a user to comprehend what actually appears on the screen when certain keys are depressed. The control commands are as follows:

Character	Explanation
<LF>	Immediately execute the LT command. This command is issued by typing the line feed key as the first keystroke after TECO's prompt. It causes TECO to move the pointer ahead one line and then type out the new line. This and the next command are useful for "walking through" a file, and examining and/or modifying lines one at a time.

<BS>	Immediately execute the -LT command. This command is issued by typing the backspace key as the first keystroke after TECO's prompt. It causes TECO to move the pointer back one line and then type the line just moved. On terminals without a backspace key, typing <CTRL/H> has the same effect.
<ESCAPE><ESCAPE>	The double <ESCAPE> character sequence tells TECO to begin execution of the command string just typed in. It echoes as $$ and also inserts two <ESCAPES> into the command string. The two ESCAPES must be typed successively to be treated as an immediate action command.
<DELETE>	Typing a DELETE character (DEL or RUBOUT on some terminals) deletes the last character typed. DELETE can be typed repeatedly to erase multiple characters. TECO echoes the deleted character whenever a DELETE is typed, to indicate that the character has been rubbed out. On a scope terminal, the character disappears from the screen and the cursor moves back one position.
<CTRL/C>	<CTRL/C> echoes as ^C (Caret-C) and aborts the entering of the command string.
<CTRL/U>	<CTRL/U> causes the current line of the current command line to be deleted. TECO echoes the character as ^U followed by <CR><LF> and an asterisk prompt. On a scope terminal, the current line physically disappears from the screen and the cursor is positioned back at the beginning of the line.
<CTRL/G><CTRL/G>	Typing two consecutive <CTRL/G> characters causes all commands which have been entered but not executed to be erased. (If the terminal has a bell, it will ring). This command is used to erase an entire command string and the two CTRL/Gs must be typed successively.
<CTRL/G><space>	<CTRL/G> followed by a space causes the line currently being entered into the command string to be retyped.
<CTRL/G>*	<CTRL/G> followed by an asterisk causes all the lines typed by the user from the last TECO prompt (the asterisk at the left margin) to be retyped.

<CR> Typing a carriage-return enters a carriage-return followed by a line feed into the command string. To enter a carriage-return without a line feed, type <CR><DELETE>.

3.5 USE OF TECO WITH RT-11

Simple TECO editing applications can utilize certain RT-11 keyboard monitor commands. Refer to Chapter 3 (Section 3) for detailed explanations of these commands.

3.5.1 Startup

Type in R TECO.

TECO is now immediately ready to accept commands.

The EDIT command EDIT/TECO filespec is used to edit an existing file. It is equivalent to -

```
R TECO
EBfilespec$Y$$
```

For RT-11 users that use TECO as the primary editor, a monitor SET command is provided -

```
SET EDITOR TECO
```

Once this command is issued, the /TECO option on the EDIT command is no longer necessary since the default editor is now TECO. As this SET command only has affect between system bootstraps, it is recommended that the command be placed in the appropriate system startup command file (eg., STARTS.COM).

If the SET command has been issued, the command EDIT filespec can be used to edit an existing file.

The standard RT-11 EDIT command options are all available with TECO:

```
EDIT/CREATE filespec
EDIT/INSPECT filespec
EDIT/OUTPUT:filespec filespec
```

3.5.2 File Specification

The file access commands ER, EB, and EW accept a file specification in the standard RT-11 format:

dev:filenam.typ

where dev: is a physical device name or a user assigned logical name. If dev: is not specified, the default DK: is assumed. The filenam field must be specified in the commands ER, EB and EW and be a legal RT-11 filename. The typ field is a file extension and must be explicitly given if used (there is no default). The EB and EW commands also accept the extended notation for an output file size;

dev:filenam.typ[n]

The optional [n] specifies the output file size where n is the number of blocks to be allocated.

3.5.3 Backup Files

The EB command maintains one level of file backup on RT-11. The pre-edited input file name is changed to:

filenam.BAK

This happens before the new output file is closed with the original name. Command EX causes this renaming to happen. If TECO is aborted or the output file is purged by the EK command, the input filename remains unchanged. Only one .BAK file for a given name is kept, as earlier .BAK backup files are deleted each time a new backup file is created.

A good policy to follow when editing, is to close the edited file frequently to ensure a minimal loss of data if an unexpected incident, eg., a power loss, should occur.

3.5.4 Re-enter and Close

The RT-11 REENTER command may always be used to continue TECO. Its primary differences from running TECO is that when REENTER is used, the text buffer and Q-register areas are unmodified. When TECO is run the text buffer and Q-register areas are cleared. The input and output file are always lost when reentering TECO. If an output file was open before reentering TECO, the file will have to be recreated with the appropriate E command. (NB: monitor commands GT ON, GT OFF, LOAD, and UNLOAD disallow a REENTER).

The output file is not closed if TECO is aborted. The RT-11 CLOSE command can be used to make the output file permanent but it will not be complete because of internal buffers that TECO keeps. TECO may be reentered after a CLOSE command.

3.5.5 File Recovery

TECO allows block replaceable devices to be opened in a non-file structured mode. This gives the user the capability to open a disk and access ASCII data anywhere on it, independent of file boundaries. The command used is -

```
ERdev:$
```

This command opens the device and then underscore or backarrow searches may be used to locate specific ASCII data and transfer it to new output files. As files tend to get reproduced, in whole or part, it is essential that the text be verified to ensure that it is complete and the correct version.

3.5.6 System Crash Recovery

If a random system failure should occur, the following procedure may help to save some or all of the editing.

1. Bootstrap the system.
2. Immediately perform a SAVE command to save as much of the memory as possible into a file on SY. The address range form of the SAVE command must be used. The SAVE command will not allow any part of the monitor to be saved.
3. Perform standard startup procedures; DATE, etc.
4. Use TECO on the SAVEd file to try and recover useful parts of the edit.

3.6 COMMON ERROR MESSAGES

If TECO encounters an erroneous command, it prints an error message and ignores this command as well as all commands which follow it. All error messages are of the form:

```
?XXX   Message
```

where XXX is an error code and the message is a description of the error. Some error messages mention the specific character or string of characters in error. In these error messages, TECO represents the non-printing special characters as follows:

Character	Form Displayed
<TAB>	<TAB>
<LF>	<LF>
<VT>	<VT>
<FF>	<FF>
<CR>	<CR>
<ESCAPE>	<ESC>
<CTRL/X>	<^X>

Every error message is followed by an asterisk at the left margin, indicating that TECO is ready to accept additional commands. If a single question mark character is typed after a TECO-generated error message, TECO will print the erroneous command string up to and including the character which caused the error message. This helps errors to be found in long command strings and determines how much of a command string was executed before the error was encountered.

A warning message is also produced by TECO. This message does not abort the command and execution continues.

% Superseding existing file

Indicates that the file to be created as the result of an EW command already exists. If the output file is closed, the old copy of the file will be deleted. The EX command can be used to effectively cancel the EW command.

These error messages are listed alphabetically by their three-letter code. This list is by no means complete; only the most common errors are described.

Code	Description
?DEV	Invalid device A file specification string in an E command contains an unknown device name.
?DTB	Delete too big An nD command has been attempted which is not contained within the current page.
?FER	File Error The file specified in an ER, EW or EB command was not found

?FNF File not found 'filespec'

The requested input file could not be located.

?FUL Output command would have overflowed output device

The page of text currently in the text buffer will not fit in the open output file. Until enough free space can be obtained in the output device, the file may have to be split. Use the EX command to close the current output file, followed by a new EW command to a temporary file. Concatenate files when space is available on the device.

?ILL Illegal command

An attempt has been made to execute an invalid TECO command.

?INP Input Error

The system has reported an error on attempting to read the current input file. The text buffer may be corrupt. This operation may be retried but if error persists, use backup file.

?NFI No file for input

Open input file by use of commands such as ER.

?NFO No file for output

Open output file by use of command such as EW.

?NRO No room for output

Output device is too full to accept output file

?OFO Output file already open

A command has been executed which tried to create an output file while an output file is currently open. Use EX command to close the output file.

?STL String too long

A search or file name string is too long. Could be the result of a missing ESCAPE after a string.

3.7 SAMPLE EDITING JOB

In this example, the entire terminal output from the editing run is reproduced intact, with reference numbers added in the left margin referring to the explanatory paragraphs which follow.

```
1< *EWSF1:DEMO.TXT$$
2< *HKIMR. D. WEBSTER
  ¦ WEBSTER ELECTRONICS PTY LTD
  ¦ BAYSWATER, VICTORIA
  ¦
  ¦ DEAR MR. WEBSTER,
  ¦
  ¦ I WAS PLEASED TO RECEIVE YOUR REQUEST FOR INFORMATION PERTAINING
  ¦ TO THE NEW TECO TEXT EDITING AND CORRECTING PROGRAM.
  ¦
  ¦ ENCLOSED IS A COPY OF THE TECO USER'S GUIDE, WHICH SHOULD ANSWER
  ¦ ALL OF YOUR QUESTIONS.
  ¦
  ¦ SINCERELY,
  ¦
  ¦
  ¦
  ¦
  ¦ $$
3< *-20LSORIA$I 3153$$
  ¦ *STION$2C13DIREGARDING$$
4< *SGUIDE$-5DIMANUAL$$
  ¦ *SELY$0T$$
  ¦ SINCERELY*OKIVERY TRULY YOURS$$
  ¦ *HT$$
  ¦ MR. D. WEBSTER
  ¦ WEBSTER ELECTRONICS PTY LTD
  ¦ BAYSWATER, VICTORIA 3153
  ¦
  ¦ DEAR MR. WEBSTER,
  ¦
  ¦ I WAS PLEASED TO RECEIVE YOUR REQUEST FOR INFORMATION
  ¦ REGARDING THE NEW TECO TEXT EDITING AND CORRECTING
  ¦ PROGRAM.
  ¦
  ¦ ENCLOSED IS A COPY OF THE TECO USER'S MANUAL, WHICH
  ¦ SHOULD ANSWER ALL OF YOUR QUESTIONS.
  ¦
  ¦ VERY TRULY YOURS,
  ¦
```

```
5< *EX$$
  (TECO is rerun using the RT-11 commands)
  *ERSF1:DEMO.TXT$EWLP:$$
6< *Y5KIMR. J. CITIZEN
  HORNBY ASSOCIATES
  1 MAIN STREET
  NEWTOWN, QUEENSLAND

  DEAR MR. CITIZEN,
  $$
  *HT$$
  MR. J. CITIZEN
  HORNBY ASSOCIATES
  1 MAIN STREET
  NEWTOWN, QUEENSLAND

  DEAR MR. CITIZEN,

  I WAS PLEASED TO RECEIVE YOUR REQUEST FOR INFORMATION
  REGARDING THE NEW TECO TEXT EDITING AND CORRECTING
  PROGRAM.

  ENCLOSED IS A COPY OF THE TECO USER'S MANUAL, WHICH
  SHOULD ANSWER ALL OF YOUR QUESTIONS.

  VERY TRULY YOURS,




  *EX$$
```

1< At this point, the user called TECO into memory. TECO responded by printing an asterisk at the left margin. The user then entered an EW command, opening an output file called "DEMO.TXT" on SF1. There is no input file. Upon receipt of the double ESCAPE ($$), TECO created the designated output file, then printed another asterisk at the left margin.

2< The user entered a command string consisting of two commands. The HK command cleared the text buffer (not a mandatory requirement, since it was already empty) and the I command inserted 18 lines of text into the buffer (includes 8 blank lines). TECO executed these commands upon receipt of the second double ESCAPE. At this point, the buffer pointer was positioned at the end of the buffer, following the last line feed character in the text.

3< The user typed -20L to move the pointer to the beginning of the buffer and SORIA$ to position the pointer immediately after the character string "ORIA" (being the last four letters of the word VICTORIA). An I command was then used to insert one space and the four-digit postal code. A second S command positioned the pointer after the word "INFORMATION". The 2C command moved the pointer to the beginning of the next line (carriage-return and line feed count two characters) and the user deleted the words "PERTAINING TO" and replaced them with the word "REGARDING".

4< The user continued editing by positioning the pointer after the word "GUIDE". This word was deleted and replaced with the word "MANUAL". Finally, a search was made for the word "SINCERELY" and an 0T command was typed to determine that the pointer was correctly positioned between the Y and the comma which follows it. An 0K command was typed to delete everything on the line except the comma. "VERY TRULY YOURS" was inserted in place of the word "SINCERELY" and an HT command caused the edited text to be printed at the terminal.

5< The command string EX$$ caused the contents of the buffer to be written onto the output file and closed the output file. The user then reentered TECO and reopened the file "DEMO.TXT" as an input file and specified the line printer as the output file.

6< This command string reads the first (and only) page of "DEMO.TXT" into the buffer, deleted the first 5 lines of text, replaced then with a different address and salutation, then printed the contents of the buffer on the terminal for verification and finally printed the new version of the letter onto the line printer. Note that the previous version of the letter still resides in file "DEMO.TXT" on SF1.

Reference:

Digital Equipment Corporation

PDP-11 TECO User's Guide	DEC-11UTECA-A-D
PDP-11 TECO User's Guide, Update 1	DEC-11-UTECA-A-DN1
TECO Pocket Guide	AV-D530A-TK

INTRODUCTION

EDIT is a line-oriented editor having character-oriented editing facilities. A student can either edit a file using simple line replacement or after gaining experience, can perform string searches/replacements, etc.

EDIT considers a file to be divided into logical units called pages with each page of text being approximately 50-60 lines in length. EDIT reads one page of text at a time from the input file into it's internal buffers where the page becomes available for editing. The editor operates in either command or text mode (albeit a special 'immediate' mode is also available for VT-11 display hardware only and will not be discussed in this chapter).

When EDIT is loaded and started (either .EDITfilespecs - at monitor level, or the SET EDIT EDIT command to set the default monitor), it is immediately in command mode; this being displayed by an asterisk at the left hand margin of the console terminal page.

Editing commands are used to locate text to be changed; to execute and verify the changes; to list an edited page on the console terminal and to output a page of text to the output file. To terminate a command, the ESCAPE key is depressed twice in succession. Execution of commands proceeds from left to right and when EDIT encounters an error before beginning the execution of a command string, it prints an error message followed by an asterisk at the beginning of the next line. This indicates that the editor is still in command mode and that execution of the illegal command has not occurred.

To enter text mode, a command is typed followed by a text string. The command entered will either insert, replace, exchange or manipulate in some form, the text. EDIT recognizes all succeeding characters as part of the text string until an ESCAPE character is initiated. This terminates the string and causes the editor to return to command mode. However, if the command requires no text, the separating ESCAPE key is not necessary. Commands are terminated by a single ESCAPE; typing a second ESCAPE, begins execution.

4.1 EDIT KEY COMMANDS

Certain special key commands are available for EDIT. Note that a control command is activated by holding down the CTRL key with the character key. The echo notation for each control/character is specified to enable a user to comprehend what actually appears on the screen when certain keys are depressed. Refer to the command demonstration displays in the 'Command Summary' Section of this chapter.

KEY	EXPLANATION
ESCAPE	Echoes as $. A single ESCAPE terminates a text string. A double ESCAPE executes the command string.
CTRL/C	Echoes at the terminal as ^C. In command mode it terminates execution and returns control to the monitor. In text mode it includes this character as part of the text.
CTRL/O	Echoes as ^O and a carriage return. Inhibits printing on the terminal until completion of the current command string. Typing a second CTRL/O resumes output.
CTRL/U	Echoes as ^U and a carriage return. Deletes all characters on the current terminal input line from the first character on the line to the entry point of CTRL/U.
RUBOUT	Echoes as \ (backslash) followed by the character deleted. Deletes a character from the current command line. Each succeeding rubout typed deletes and echoes another character. An enclosing backslash prints when a key other than RUBOUT is typed. This erasure is done from right to left. Since EDIT accepts multiple-line commands, RUBOUT can delete past the carriage return/line feed combination and delete characters on the previous line. RUBOUT can be used in both command and text mode. The DELETE key has the same effect as the RUBOUT key; it depends on the keyboard used.
TAB	Spaces to the next Tab stop. Tab stops are positioned every eight spaces on the terminal; pressing the TAB key causes the carriage to advance to the next tab position.
CTRL/X	Echoes as ^X and a carriage return. Causes the editor to ignore the entire command string being entered. Editor reverts to the starting command asterisk.

4.2 EDIT CHARACTERISTICS

Command strings can be used with spaces, carriage returns and line feeds as EDIT ignores them. It improves the readibility of the command. However EDIT does utilize these features when inserting text strings.

Most EDIT commands function in conjunction with a movable reference pointer. This pointer precedes the first character in the buffer and is always located between two characters and never directly on a character.

EDIT commands are either character or line oriented. Line terminating characters determine the line count on commands.

Portions of command strings can be executed more than once by enclosing the portion in <angle brackets> and by preceding the left angle bracket with the number of iterations.

The memory area used by the editor is divided into four logical buffers:

the text buffer contains the current page being edited;

The command input buffer holds the command that is currently being typed at the keyboard;

the save buffer holds stored text;

the Macro buffer contains the command string entered with the Macro command.

With the latter two buffers, EDIT does not allocate space until required and returns space to the free area of memory when the reverting commands are executed.

4.3 EDIT COMMAND STRUCTURE

The general syntax for all the EDIT commands is -

[n]C[text]$ or [n]C$

where -

n represents an argument which may be either numerical or textual and is positioned before a command letter. It specifies either the number of times the command is to be performed or a particular portion of text to be affected by the command. Not all commands require an argument and some commands can be preceded by either a positive or negative numeric value. The positive (unsigned) command will move a pointer a number of times forward (L to R) and a negative (-) command will revert the pointer backward.

C represents a one or two letter command (refer to the Command Summary).

text represents a string of ASCII characters.

NOTE: Other arguments are in the form -

0 Indicates the text between the beginning of the current line and the reference pointer;

/ Refers to the text between the reference pointer and the end of text in the buffer;

= Represents -n, where n is equal to the length of the last text argument used. This argument may only be used with the J, D, and C commands.

EDIT commands fall into seven general categories -

This table is designed to serve as a quick reference for the command codes; more explicit details are available in the summary.

CATEGORY	COMMANDS	CODE	SYNTAX OF THE COMMAND
File open and close	Edit Backup	EB	EBdev:filnam.typ[n]$
	Edit Read	ER	ERdev:filnam.typ$
	Edit Write	EW	EWdev:filnam.typ[n]$
	End File	EF	EF
File input/output	EXit	EX	EX
	Next	N	N
	Read	R	R
	Write	W	nW
Pointer location	Advance	A	nA
	Beginning	B	B
	Jump	J	nJ

Search	Find	F	nFtext$
	Get	G	nGtext$
	Position	P	nPtext$
Text listing	List	L	nL
Text modification	Change	C	nCtext$
	Delete	D	nD
	eXchange	X	nXtext
	Insert	I	Itext$
	Kill	K	nK
Utility	Edit Console	EC	EC
	Edit Display	ED	ED
	Edit Lower	EL	EL
	Edit Upper	EU	EU
	Edit Version	EV	EV
	Execute Macro	EM	nEM
	Macro	M/c.s.	M/command string$
	Save	S	nS
	Unsave	U	U

NOTE:

The default used in the explanation of the syntax command is n. The meaning of other syntax command abbreviations is as follows -

dev:

represents either a logical device name or a physical device name. If not specified the logical device name defaults to SY: and the physical device name defaults to DK:

filnam.typ

is the name of a file. No spaces or tabs are allowed in the file name or type and as many as three output and six input files are allowed. If the dot and file type is omitted, the system may apply a default file type that the program specifies.

[n]

is an optional declaration of the number of blocks (n) required for the output file. n is a decimal number (up to 65535) which must be enclosed in square brackets.

4.4 EDIT COMMAND SUMMARY

Where arguments are applicable, these are included in the command summary.

COMMAND	EXPLANATION
File Open and Close	
EB	Opens an existing file for editing and at the same time creates a backup version of the file. No text is read or written with this command.
ER	Opens an existing file for input and prepares it for editing. Only one file at a time can be open for input.
EW	Opens a file for output of newly created or edited text. Only one file can be open for output at a time.
EF	Closes the current output file and makes it permanent.
File Input/Output	
EX	Terminates the editing session and returns control to monitor. It writes the text buffer to the output files, transfers the remainder of the input file to the output file, closes all open files and renames the backup file with a .BAK file type if an EB command is in effect.
nN	Executes the N command n times. N writes the contents of the text buffer to the output file, deletes text from the buffer and reads the next page of input file into the buffer.
R	Reads text from an input file into the buffer.
nW	Writes n lines of text beginning at the pointer and ending with the nth end-of-line character to the output file.
-nW	Writes n lines of text to the output file beginning with the first character on the -nth line and terminating at the pointer.
0W	Writes to the output file the current line up to the pointer.

Pointer Location

nA	Moves the pointer forward n lines and positions it at the beginning of the nth line.
-nA	Moves the pointer backward past n carriage return/line feed combinations and positions it at the beginning of the -nth line.
0A	Moves the pointer to the beginning of the current line.
/A	Moves the pointer to the end of the text buffer.
B	Moves the current location of the pointer to the beginning of the text buffer.
nJ	Moves the pointer forward n characters.
-nJ	Moves the pointer backwards -n characters.
0J	Moves the pointer to the beginning of the current line.
/J	Moves the pointer backward n characters where n equals the length of the last text argument used.

Search

nF	Starts at the current pointer location and searches the entire input file for the nth occurrence of the text string.
nG	Searches the current text buffer for the nth occurrence of a specific text string, starting at the current location of the pointer.
nP	Deletes the contents of the buffer after it has searched without writing any text to the output file.

Text Listing

nL	Prints at the terminal n lines beginning at the pointer and ending with the nth end-of-line character.
-nL	Prints all characters beginning with the first character on the -nth line and terminating at the pointer.

0L	Prints the current line up to the pointer. This command is used to locate the pointer within the line.
/L	Prints the text between the pointer and the end of the buffer.
V	Prints at the terminal, the entire line in which the pointer is located.

Text Modification

nC	Replaces n characters following the pointer with the specified text. Places the pointer after the inserted text.
-nC	Replaces n characters preceding the pointer with the specified text. Places the pointer after the inserted text.
0C	Replaces the current line up to the pointer, with the specified text. Places the pointer after the inserted text.
/C	Replaces the text beginning at the pointer and ending with the last character in the buffer. Places the pointer after the inserted text.
=C	Replaces -n characters with the indicated text string, where n represents the length of the last text argument used.
nD	Deletes n characters following the pointer. Places the pointer before the first character following the deleted text.
-nD	Deletes n characters preceding the pointer. Places the pointer before the first character following the deleted text.
0D	Deletes the current line up to the pointer. The position of the pointer does not change.
/D	Deletes the text between the pointer and the end of the buffer. Positions the pointer at the end of the buffer.
=D	Deletes -n characters, where n equals the length of the last text argument used.

nX	Replaces n lines, including the carriage return and line feed characters following the pointer. Places the pointer after the inserted text.
-nX	Replaces n lines, including the carriage return and line feed characters preceding the pointer. Positions the pointer after the inserted text.
0X	Replaces the current line up to the pointer with the specified text. Positions the pointer after the inserted text.
/X	Replaces the text beginning at the pointer and ending with the last character in the buffer with the specified text. Positions the pointer after the inserted text.
I	Is the basic command for inserting text.
nK	Removes the character string (including the carriage return and line feed combination) beginning at the pointer and ending at the nth end-of-line.
-nK	Removes the character string beginning at the nth end-of-line preceding the pointer and ending at the pointer. Thus if the pointer is at the centre of a line, the modifier -1 deletes one and one-half lines preceding it.
0K	Removes the current line up to the pointer.
/K	Removes the characters beginning at the pointer and ending with the last line in the text buffer.

Utility

EC	If a scroller is in use, a window into the text buffer appears in the top two-thirds of the screen, while the bottom third displays command lines. The EC command is used to revert the scroller to its normal mode so that text and commands use the full screen.
ED	Use this command to recall the window.
EL	When issuing an EL command, it means that editing can be actioned in upper or lower case. When starting up EDIT, upper case is assumed. If using EX or CTRL/C, the EDIT will default back to upper case. Note also that text specified in search commands must be specified in the proper case.

EU	Use this command to revert back to upper case if EX or CTRL/C are not to be used.
EV	This command is for information purposes only; it displays the version number of the editor in use at the console.
nEM	This command executes the command string previously stored in the macro buffer by the M command.
M/c.s.	This command inserts a command string (c.s.) into the EDIT macro buffer. The / (slash) represents the delimiter character. The delimiter is always the first character following the M command and can be any character that does not appear in the macro command string itself. The macro command does not affect the contents of the text or save buffers.
0M or M//	Clears the macro buffer and reclaims the area for text.
nS	Lets text be stored in an external buffer called a save buffer and subsequently it can be inserted in several places in the text. The save command copies n lines, beginning at the pointer in a forward direction (a negative argument is not allowed). Previous contents of the buffer are destroyed.
U	Inserts the contents of the Save buffer into the text buffer (repeat performance applicable).
0U	Clears the save buffer and reclaims the area for text.

4.5 GENERAL USE OF EDIT

It is recommended that text be inserted in small sections rather than all at once. This could eventually be a time-saving operation and would save considerable re-entering in the case of power failure or human error. The procedure taken when entering large volumes of text is as follows:

1. Open the file with an EB command;
2. Insert or edit a few pages of text;
3. Insert a unique text string, eg., \|\|\|, to mark the place;
4. Use the Exit command to secure the editing to date;

5. Start again, using the F command to search for this unique string;

6. Delete this marker string (using DELETE or RUBOUT) and continue editing.

4.6 COMMON ERROR MESSAGES

Listed are some common error messages that may occur -

?EDIT-F-Command aborted
A command was prematurely terminated because CTRL/C was typed twice.

?EDIT-F-Command buffer full;no command(s) executed
If possible, empty the save or macro buffers, or write part of the text to the output file. Retype the command(s) as a series of smaller commands or shorter sequences.

?EDIT-F-End of input file
An attempt has been made on a READ, NEXT or file search command but no input data was available because the end of the file was reached. Close the file by using EX or EF; reopen it if more editing remains to be done.

?EDIT-F-File not found
Check for a typing error in the command line. Verify that the file name exists as entered in the command line and retry the operation.

?EDIT-F-Illegal file name
Illegal file name specified in an EB, EW or ER command or as input to the EDIT command. Also check that the dev:filnam.typ[n] specification does not exceed 19 characters and is in the proper format.

?EDIT-F-Insufficient memory
An attempt was made to use the I, S, U, R, N, C, or E commands when there was not enough room for the appropriate buffer. Delete unwanted buffers to create more room by using the OU or OM commands. EDIT does not allocate space for the macro and save buffers until an M or S command executes. Only when an OM or OU command (Unsave) is issued will the space be returned to the free area.

?EDIT-F-Output file full
Close the file using the EF command or use the EW command to close the file and open a new file for output.

?EDIT-W-Command buffer almost full
The command entered is within 10 characters of exceeding the space available in the Command Buffer. Complete the command using less than 10 characters or type ESCAPE twice to execute that portion of the command line already completed and then enter the remainder as a second command.

?EDIT-W-Superseding existing file
An EW or EB command was issued for an already existing file name. If the new file is closed, the old one will be deleted. To avoid replacing the file, terminate the edit with CTRL/C, followed by two ESCAPES and then the monitor REENTER command.

Reference:

Digital Equipment Corporation
RT-11 System User's Guide
AA-5279B-TC

INTRODUCTION

The Floppy disk utility FUTIL is a diagnostic maintenance program. It provides the user with a convenient means to create various formats on floppy disks, to copy a disk, to list bad blocks on a disk and to compare two disks. While performing any of these functions, the system will display any soft (recoverable) errors or hard (non-recoverable) errors detected. A list of possible error messages is specified at the end of this chapter.

This program is more dangerous to use than BACKUP for backing up data disks, because no inherent protection is afforded against inadvertent backwards copy of the blank disk over the desired data, giving a net result of two blank disks!

If FUTIL is driven from a .COM file, any hard errors or data errors discovered will terminate the .COM file execution on exit from FUTIL. Monitor swapping has been avoided on FUTIL exit to RT-11, allowing the system drive to be left write-protected without getting swap-error messages.

FUTIL cannot run under TSX or TSX-Plus.

Using FUTIL

Enter the RT-11 Operating System and type in R FUTIL.

The system displays:

```
Spectrum FUTIL Version 6
Floppy Disk Utility

     0    Format & Verify
     1    Copy and Verify
     2    List bad blocks
     3    Compare 2 disks
     9    Return to RT11

     ACTION:
```

Select the number of the option required.

Format and Verify

It is usual to insert the system disk in Drive 0 and the blank disk in Drive 1. These instructions will relate as such.

By selecting Option 0, a menu of the various formats that may be used is displayed. Check the specifications of the model number (Spectrum Eleven Hardware Manual) and establish the native format type for the machine. Other formats which may be selected are also listed in this section under the Additional Format Capability.

The formats 1S, 1D and 2D have strictly sequential sector addressing, with track-to-track spiralling of the sector zero reference in order to optimize long sequential transfer rates. This applies particularly when multi-block transfers are called from the application, such as occurs with most standard utilities and operating system functions.

The timing of intersector gaps is 1 or 2 milliseconds. This means that when an application calls sequential transfers as a series of one block requests, with some inter-block processing, there is in general insufficient time to catch the next sector without losing a complete revolution (166 milliseconds). The result can be an 8:1 or 16:1 degradation of sequential access performance.

The formats 1SI, 1DI, and 2DI have an interleaved sector addressing scheme which allows 10 or 20 milliseconds between sequential sector addresses, which in some cases results in a profound improvement in single block sequential access. However this results in 2:1 degradation in multiblock sequential access performance.

Random access performance is unaffected by choice of interleaved or non-interleaved formats.

Enter OPTION: 0

The system displays:

Target drive (0-3): 1

(Enter the number of the drive containing the disk to be formatted and WRITE PROTECT the other drive/s).

The system displays:

```
Non interleaved formats:
0  DX   IBM/DEC Interchange format
1  1S   Spectrum single head/density
2  1D   Spectrum 1 head dual density
3  2D   Spectrum 2 head dual density
Interleaved formats:
4  1SI  Spectrum single head/density
5  1DI  Spectrum 1 head dual density
6  2DI  Spectrum 2 head dual density

Selection:
```

Enter the number of the option required followed by the RETURN key. If the disk is to be used in an application where there is a fair amount of computation involved between sequential reads, it is recommended that an interleaved format be selected to improve performance.

The system displays:

WARNING! Previous data will be lost, type 0 to proceed.

Enter 0 (RETURN key)

When formatting is complete the system displays:

Testing ....

This indicates that a read-verify of the formatted disk is in progress.

The system displays:

Operation successful

and re-displays the Menu options.

If the operation was not successful, refer to the Error Message paragraph at the end of this chapter.

Copy and Verify

A copy and verify process will copy the entire source disk to another disk. A blank (unused) disk must be formatted prior to this copy function. Assuming a copy is to be made after a disk has been formatted, it is not necessary to re-enter FUTIL. Simply remove the system disk from Drive 0 and replace it with the source disk. The formatted disk is in Drive 1. If the target disk being used is a disk containing unrequired/obsolete data, it has obviously been formatted previously, and reformatting is not necessary. Extreme care must be taken that the disk is definitely not required as all existing data will be overwritten.

On single drive Spectrums, or in cases where the second drive is faulty, FUTIL allows selection of the same drive as Source and Target for a copy operation, and the user is prompted to alternate the two diskettes while the transfer is affected one core-full at a time.

Enter OPTION: 1

The system displays:

```
Install Source Disk
Source Drive (0-3): 0
```

(Enter the number of the drive that contains the source disk and WRITE PROTECT this drive).

The system displays:

```
Source Format is NN
```

where NN is the sensed format type of the disk. (If the incorrect format type has been selected, enter ^ up arrow or circumflex to return to initial menu and commence again).

```
Install target disk
Target drive (0-3): 1
```

(Enter the number of the drive that contains the formatted disk which will become a copy of the disk in the Source Drive. Note that for a single drive copy operation, the Target and Source drives will be identical.)

The system displays:

```
Target format is NN
WARNING! Previous data will be lost.
Type 0 to proceed:
```

If at this point it is realized that an error has been made, enter up-arrow or circumflex to return to initial menu and start again.

Before typing 0 to proceed, check that WRITE PROTECT has been activated on the source drive.

NB: Activation of the WRITE PROTECT button is not mandatory, it is merely a security precaution in case the incorrect drive numbers have been entered. If this is the case, a message will display repeatedly to warn the operator that an attempt is being made to overwrite a WRITE PROTECTED disk. To rectify this error, enter a CTRL/C combination and start the copy/verify process again.

On completion of the copy process, the system will display:

```
Testing ...
```

The two disks are compared to ensure that the information is identical.

The system then displays:

Operation successful

and re-displays the Menu Options.

If the operation was not successful, a 'data compare error' message is displayed.

NB: If the formats are identical, the program will make a straight one-for-one copy. If the formats are different, only the capacity of the smaller of the two will be transferred and the data will be re-arranged to suit. This provides one means of translation of IBM/DEC universal interchange disks to the various WEBSTER formats.

List Bad Blocks

Enter OPTION: 2

The system displays:

Install Target Disk
Target drive (0-3):

(Enter the number of the drive containing the disk that requires the bad blocks to be listed).

The system displays:

Target format is NN
Testing ...

Where no bad blocks are present, the system displays:

Operation Successful

and then re-displays the Menu Options.

Where bad blocks are present, the system displays all pertinent details, ie., Drive, Block, Head, Track, Sector and whether or not the error is recoverable (soft error) or unrecoverable (hard error). A summary of the number of each type of error is also displayed.

Compare 2 Disks

This operation which is same procedure used in the testing phase of the copy operation, verifies that two disks are identical, or establishes the location and nature of differences.

For a data compare error FUTIL will report the block number, word number, and the two octal data images which miscompare. Only the first miscompare of each block is reported.

Enter OPTION: 3

The system displays:

Install Source disc
Source drive (0-3):

(Enter the number of the source disk drive).

The system then displays the source disk's format.

The system displays:

Install Target disc
Target drive (0-3):

(Enter the number of the target disk drive)

The system then displays the target disk's format.

Testing ....

On completion the system displays:

Operation Successful (if the two disks compared)

and then re-displays the menu options.

If the comparison was not successful, the system displays the block, the error message 'Data Compare Error', the source data address, the target data address and the word number.

Return to RT11

If one of the above options has been actioned and no further option is to be selected, it is necessary to return control to the system monitor.

Ensure that the system disk has been re-installed in Drive 0.

Enter OPTION: 9

The system returns control to the monitor and displays the monitor prompt.

ERROR MESSAGES

The following two messages may occur in any option from 0 to 3.

Target is not Recognised

Drive not On Line

If either of these messages occur:

Check that the disk has been inserted correctly, ie., face up with the label nearest the drive opening;

Check for an alien format;

Check Source Target disk is formatted;

Incorrect format for drive type.

Record Not Found

This message signifies that the header information for a specific track or sector cannot be found.

No double head capability

The machine does not have a drive with two heads. The incorrect format type has been selected, check the specifications for the Spectrum model (Spectrum Eleven Hardware Manual).

No double density capability

The machine cannot create double density format. Check the specifications of the Spectrum model.

Data compare error

FUTIL reports the first miscompare of each block and displays the block number, word number, and the two octal data images which miscompare.

?FUTIL-F-Foreground Job Present

FUTIL can only operate as a single job under the SJ monitor or as the background job of the FB monitor with no job present in the foreground.

?FUTIL-F-Insufficient buffer space

There is insufficient memory available for FUTIL to operate.

INTRODUCTION

BACKUP enables a user to save the full contents of a floppy disk and if necessary, to recover these contents. This program has been introduced to make it difficult to accidently overwrite required disks. This is established in two ways:

1. By having the ability to include a disk identification and owner in the 'home block';

2. By having two types of disk, 'USER' the current source disk, and 'BACKUP' the target disk onto which User disks are copied. This 'BACKUP' type of disk is made unusable by any other program (this is done by corrupting the disk's Directory).

Note that in the procedures described, the System Disk may be removed as soon as BACKUP is called. The program does not use the System Disk again until the 'Return to RT11' option is used. Also, at any point in the BACKUP dialogue, CTRL/C will return the user to the option menu.

The first time BACKUP is run, an identification must be issued to the target disk. Assuming a brand new disk is being used as the target disk, first format it using FUTIL.

Using BACKUP

Enter the RT-11 Operating System and type in R BACKUP

The system displays:
Spectrum BACKUP Version 2
Floppy Disk Backup Utility

Select one of the following options:-

0. Backup a disk
1. Recover a disk
2. Display a Disk's i.d.
3. Change a Disk's i.d.
4. Return to RT11

OPTION:

NB: As stated if this is the first time this disk has been used for BACKUP, select Option 3 first to create an identification, ie., this will be a change from a blank field to the required identification.

The options will be discussed in the following sequence -

2. Display a Disk's i.d.
3. Change a Disk's i.d.
0. Backup a disk
1. Recover a disk
4. Return to RT11

Display a Disk's i.d.

Enter OPTION: 2

The system displays:
DISK [0-3]?

Enter the number of the disk drive that holds the floppy disk requiring identification.

The system displays:

```
Disk Details:-
BACKUP Copy Date:
Volume i.d.     :
Owner Name      :
Disk Type       :
Disk i.d.       :
```

where the following applies -

BACKUP Copy Date	A date will only appear if a recovery has ever been actioned off a BACKUP disk;
Volume i.d.	Is the identification of the file type;
Owner Name	The name allocated for ownership;
Disk Type	Will be USER. When BACKUP is processed, the backup disk will automatically specify BACKUP and append 'Backup' adjacent to the Volume i.d.;
Disk i.d.	The name given to the job being processed if a data disk or system disk.

The system also displays the available options again so that a further selection may be made. If no further option is required, return to RT11 (option 4).

Change a Disk's i.d.

Enter OPTION: 3

The system displays:
DISK [0-3]?

Enter the number of the disk drive that holds the floppy disk requiring a changed or implemented identification.

The system displays:

```
Disk Details:-
BACKUP Copy Date      :
Volume i.d.           :
Owner Name            :
Enter New Owner       :
```

If no change is required, enter RETURN key to accept the display.

The system then displays:

Disk Type :

followed by the message -

Do you wish to change the disk's type [Y/N]?

(It is not necessary to enter the response in upper case.)

If No, enter N (RETURN key)

If Yes, enter Y (RETURN key) and the system displays:

Make the disk BACKUP type [Y/N]?

(The type should either be BACKUP or USER. If BACKUP type is accepted then then the Volume i.d. will change with the word 'Backup' displayed next to the Operating System designation, ie., RT11ABackup, and the Disk Type will then display **BACKUP**.)

The system displays:

```
Disk i.d.      :
Enter new i.d. :
```

If no change is required, enter RETURN key to accept the displayed field.

The system then displays the original options. If no further option is required, return to RT11 (Option 4).

Backup a Disk

Enter OPTION: 0

The system displays:
*** Disk BACKUP ***

Place the disk to be backed up (Source Disk) in Drive 1 and activate the WRITE PROTECT button. Place the backup disk (target/output disk) in Drive 0.

Enter 'Return' when ready - enter RETURN key.

The system displays:

Volume i.d.
Owner Name
Disk Type
Disk i.d.

Possible Error Messages:

If the Owner Name and/ or Disk i.d. do not match, the system will 'bleep' and display an appropriate message. This will always happen if a newly formatted disk has not been processed through Option 3. Check the i.d. of both disks. If necessary amend the Owner Name and/or i.d. - assuming that they are the correct disks.
Extreme care must be taken when labelling floppy disks. The incorrect disk may be loaded (source or output disk).

On completion of the backup routine, the system displays the original options menu. Insert the systems disk in Drive 0 to return to the RT11 Operating System (Option 4).

Recover a Disk

Enter OPTION: 1

The system displays:
*** Disk RECOVERY ***

Place the Backup Disk (Source Disk) in Drive 0. Place new formatted disk (will become the recovered information disk) in Drive 1. Press the WRITE PROTECT button on Drive 0. Enter 'RETURN' when ready.

The system displays:

Disk i.d. is :
Volume i.d. :
Owner Name :
Disk Type :
Disk i.d. :

After the job is complete, the system displays:

** Operation Complete ** - followed by the Menu options.

To verify that all the recovery phases have been carried out correctly, select Option 2 to display the i.d. of the newly created disk.

The fields should read as follows:

```
Disk Type      : USER
Owner Name     : )     must be identical to the BACKUP disk.
Disk i.d.      : )
```

The BACKUP copy date field will display the machine date entered during bootstrap.

Return to RT11

Enter OPTION: 4

The system displays the message -

Ensure System Disk is mounted in Drive 0, enter 'RETURN'.

Control is passed to the system monitor.

INTRODUCTION

The BAKFLP Utility is designed to save selected large files from a large disk memory such as a Winchester disk, to a series of formatted and initialized floppy disks. If the size of a file to be saved does not exceed 2432 blocks (approximate number of available blocks on a floppy disk), use the RT-11 COPY command. A further option is also available to reinstate these files back to the main storage device. The limitation on the number of files to be saved and/or restored is 72 with no restrictions on the number of diskettes used. As a guide, it would take approximately 24 diskettes to hold the contents of a Model G (29.5 megabyte) Winchester disk.

BAKFLP offers the user protection against overwriting wanted floppy disks in that it requests volume identifications. A directory of the selected files is written to Volume 1 of the floppy disk set. The files are then copied in the sequence entered. It is irrelevant if a file overflows the storage area on a floppy disk as the system prompts for the next consecutive numbered volume to be installed, and then continues to save the rest of that file. When a file is to be restored, BAKFLP reads the directory on Volume 1, calculates where the file is located and then prompts the user to install the correct volume. File protected status is preserved in the Save operation.

Note: For copying a whole disk, use SAVRES.

PREPARATIONS PRIOR TO USING BAKFLP

1. Allocate sufficient diskettes and check that all have been formatted using the Webster floppy disk utility FUTIL.

2. Initialize each diskette with the RT-11 INITIALIZE/VOLUMEID command. For example, to initialize the floppy disk in unit SF0: the command is -

   INIT/VOL SF0:

   The system will respond with a warning message (only if running under TSX/TSX-Plus) and then display -

   SF0:/Initialize; Are you sure?

   If an error has been made and this command is not required, enter an N to abort the command. Otherwise enter a Y and the system will display -

   Volume ID?

Each floppy disk is to be allocated a number which is entered at this point. The volumes are numbered consecutively, beginning at one.

The system then displays -

Owner?

Respond by depressing the RETURN key. At the start of the Save operation the system date is written into the owner area of the Volume Id.

When each diskette is initialized, it should be labelled with the job name and it's volume number. Eg., BAKFLP VOLUME 1, BAKFLP VOLUME 2, etc.

3. Prepare a list of the files to be either saved or restored. Consider all transaction, edit, master, and virtual unit files that have been accessed and updated since the last backup session. It is not necessary to save any system or program files as these do not normally alter from one session to the next.

4. When running the Save procedure, use the RT-11 keyboard monitor ASSIGN command and enter -

```
ASSIGN dev1 SAV
```

where dev1 is the floppy disk unit. For example -

```
ASSIGN SF0 SAV
```

When running the Restore procedure an asssignment must also be made for RES, the device to which the files are to be restored.

```
ASSIGN dev1 SAV
ASSIGN dev2 RES
```

where dev2 is any file structured device other than dev1. For example,

```
ASSIGN SF0 SAV
ASSIGN DM0 RES
```

this assignment tells BAKFLP to restore from SAV (SF0) to RES (DM0).

CONSIDERATIONS AND GENERAL RULES

1. To ensure a completely error free backup of a file it is advisable although not mandatory to request a read-after-write check for every record written to the file.

   Use the RT-11 keyboard monitor command -

   SET SF RAW

   The system retries if an output error occurs. If three retries fail, the system indicates an output error -

   ?BAKFLP-F-Error Writing File (filespec)

   In all probability the problem lies with the floppy disk so the only recourse is to replace the disk with another floppy which has been formatted, and initialized with the same Volume Id to ensure continuity of volume numbers.

   The normal system mode is NORAW which writes every record directly without reading it back for verification. This significantly increases transfer rates at the slight risk of increased error rates.

2. When restoring files, ensure that there is sufficient space on the restore device. Enter DIR/FULL to check the number of contiguous blocks available on device DK. If necessary, deleted unwanted files and perform a SQUEEZE operation. Where there is not sufficient space to restore the files, the message 'Cant open output file' is displayed.

3. Activating the WRITE PROTECT button on the read device would provide further security. However under TSX/TSX-Plus, care must be taken to ensure that other users are not updating programs that write to disk. Certain error conditions occur if the wrong device is write protected.

Save Operations from Winchester to Floppy Disk

If SF is Write Protected at the start of the Save operation, the program will only abort after the file names have been entered. The message displayed is -

?BAKFLP-F-Error Writing Directory

Restore Operation from Floppy to Winchester Disk

SF is Write Protected

If DM is Write Protected the program will abort after the file names have been entered and an RT-11 message is displayed.

?MON-F-Directory I.O Error

4. During a Restore operation, files should be restored in the same order in which they were saved to eliminate frequent volume changing.

5. An Indirect Command File may be set up to incorporate many file names. It is conventional to use a .COM file type for an indirect file.

6. To abort BAKFLP at any time, type in CTRL/C (hold down CTRL while depressing C).

USING BAKFLP

Type R BAKFLP.

The system displays -

BAKFLP Version 1.01
HELP,SAVE or RESTORE [H/S/R]?

Select one of the above options.

HELP

Type in H.

The HELP command describes the BAKFLP Utility and briefly explains the more important user functions.

SAVE

Type in S and load Volume 1.

If the device assignment has not been made prior to entering R BAKFLP, the system displays -

Please assign SAV to a floppy disk unit and try again.

Against the period, type in the required ASSIGN command followed by the RETURN key. Re-enter R BAKFLP and select option S.

The system displays an asterisk (*) prompt. Enter the filenames terminated by the RETURN key. The default source storage device is DK. Continue in this manner until all required filenames have been entered. A filename which is not valid will cause the message 'The input file (filespec) was not found' to display. To terminate the input operation type RETURN with no preceding filename.

The system now has a complete directory image of the files to be saved, and a corresponding list of their devices.

If at this point no volume is loaded in device SAV, the system displays -

```
?BAKFLP-W-Error Reading home block
Mount Volume 1 in 'dev1' and type the RETURN key to continue.
```

The program will continue to prompt the user for the next consecutive volume number until all files are saved. If an incorrect Volume is loaded the system displays -

```
You have mounted the wrong volume.
Please mount Volume 'n' in 'dev1' and type the RETURN key to
continue.
```

The Save operation is terminated when 'Save Complete' is displayed.

RESTORE

Type in the option R and load Volume 1.

If the assignment to device SAV has not been made prior to entering R BAKFLP, the system displays -

```
Please assign SAV to a floppy disk unit and try again.
```

If the assignment to device RES has not been made prior to entering the filename(s), the system displays -

```
Please assign RES to a restore device and try again.
```

If an incorrect volume is already loaded, the system displays -

```
Mount Volume 1 in 'dev1' and type the RETURN key to continue.
```

If no volume has been loaded, the system displays -

```
?BAKFLP-W-Error reading home block.
Mount Volume 1 in 'dev1' and type RETURN to continue.
```

The system displays an asterisk (*) prompt. Enter the filenames terminated by the RETURN key. Continue in this manner until all required files have been entered. A filename which is not valid will cause the message 'The input file (filespec) was not found' to display. To terminate the input operation type RETURN with no preceding filename.

The program then reads the Directory on Volume 1 and stores the filenames in a buffer. The user is then prompted for the required volume by its Volume Id.

If an incorrect Volume is loaded, the system displays -

> You have mounted the wrong volume.
> Please mount Volume 'n' in 'dev1' and type the RETURN key to continue.

The Restore operation is terminated when 'Restore Complete' is displayed.

INTRODUCTION

SAVRES is a save/restore utility designed for use on Spectrum Winchester (RK06/07 compatible) disks when configured with a floppy disk sub-system or magtape (TM-11 compatible) sub-system. SAVRES must therefore always have DM (Winchester) as one of the devices with either SF (Floppy) or MT (magtape) as the other device.

SAVRES performs an image copy from the input device to the output device. If a specific file(s) is to be saved, use BAKFLP. Specific files can only be saved on magtape using the RT-11 COPY command.

SAVRES can operate under RT-11SJ or TSX-Plus. Under RT-11, magtape save/restore running time is around 3 minutes (plus any tape load and rewind time) and floppy disk time is approximately 50 seconds for each disk (which includes disk swapping and handling time). For TSX-Plus, the time taken varies depending on system loading.

No directories are written on volumes although volume identification is required on floppy disks. SAVRES caters for file overflow from one floppy disk to another, and for magtape uses 8K blocks giving a capacity of just over 40 Mb per tape which makes it convenient to use one 2400' tape per 'DM' unit (28-33 Mb).

PREPARATIONS PRIOR TO USING SAVRES

Magtape routine

1. Allocate sufficient tapes according to the number of units per Winchester disk:

Model	Units	No. of Tapes
GCM	DM0:	1
KM	DM0:	
	DM1:	2
JM	DM0:	
	DM1:	
	DM2:	
	DM3:	4

2. If a Save operation is being performed, prepare a tape label with date, time, and any other pertinent information and insert a write ring in the tape.

3. If a restore operation is being performed, ensure that the correct tape has been selected and check that the write ring has been removed.

Floppy Disk routine

1. Allocate sufficient floppy disks to hold the contents of the Winchester disk.

Model	Size	No. of Diskettes
HC	8.5 Mb	7
GC	29.5 Mb	24

2. If the diskettes have not been previously formatted, format using FUTIL.

3. Initialize each floppy disk using the RT-11 INIT/VOL command. Each disk requires an identification with the Volume number commencing from 1 for the first disk, and incrementing by one for each subsequent disk. Simply enter the RETURN key for 'Owner' as this field is not required by SAVRES. As each disk is initialized it should be labelled with it's volume identification number.

   Note: Once a set of floppy disks is initialized with INIT/VOL, it is not necessary to repeat this procedure. However it is always wise to have two sets of disks and back up to the alternate set each time.

CONSIDERATIONS AND GENERAL RULES

1. SAVRES uses standard CSI format for all commands. When SAVRES runs, it prompts with an asterisk (*) and requires a CSI equation to specify input and output devices.

   The CSI equation also accepts 3 optional switches of which the latter two are not normally used.

   /Q (QUIT) Returns to RT-11.
   /B:n Specifies beginning block on DM unit
   /E:n Specifies ending block on the DM unit

2. To exit from SAVRES at any point in the procedure, type CTRL/C (twice in succession).

3. After each response is typed, depress the RETURN key.

4. Activating the WRITE PROTECT button on the read device would provide additional security. However, under TSX-Plus, care must be taken to ensure that other users are not updating programs that write to disk. Certain error conditions occur if the wrong device is write protected.

5. The system must be bootstrapped from either the input or output device. Ie., it is not permissible to boot onto floppy and then attempt to restore from magtape to Winchester disk. If a Winchester is corrupted and it is to be restored from magtape, either boot from magtape or create a bootable system on the Winchester and boot from that (refer Magtape Bootstrap Procedures, Section 6).

Using SAVRES for Winchester/Magtape Operation

These instructions are for a Save operation and assume running under RT-11. Where relevant, details for a Restore procedure are documented in addition to the Save details.

Loading Tape

Insert write ring and load the tape (ring down).
Press LOAD REWIND and then ON-LINE.

The two indicators will light up and flicker as the tape loads automatically. When the load operation is complete, these indicator lights remain stationary and the WRT EN indicator illuminates (Note: During restore, which only reads the tape, this indicator will not light up).

Running SAVRES

Enter R SAVRES

The system displays -

```
SAVRES - V03.00
*
```

The user now enters a CSI command string in the format:

dvn:*=DVn:

where dvn (output device) is the device to be replaced by DVn (input device). The output device is always the device specified before the * and the input device is always the device specified after the = sign. For example:

MT4:*=DM0:

This command saves from Winchester disk to magtape using streaming mode. The streaming mode (MT4:) is normally used during SAVRES as it is a lot faster than the start/stop mode (MT0:).

DM0:*=MT4:

This command restores from magtape to the Winchester disk.

Enter the RETURN key to start the save procedure.

For a Save operation the system displays -

Save DM0: onto MT4: are you sure?

For a restore operation, the system displays -

Restore DM0: from MT4: are you sure?

If you type N (No), the system displays -

?SAVRES-F-Operation unsuccessful
. (monitor prompt)

If you type Y (Yes), the system displays -

Savres tape must now be loaded. Hit RETURN to continue.

Load the tape if not previously loaded. If the RETURN key has been pressed and the tape is not loaded or not ON-LINE, the system displays the following message:

?MON-F-Directory I/O error.

Commence SAVRES from the beginning.

When SAVRES is running, the ACCESS light on the Winchester will pulsate and the LOAD light on the tape unit will extinguish. The system then displays the blocks in consecutive-thousand numbers, eg.,

Block: 1000
Block: 2000
Block: 3000
etc

On completion of the save operation the system displays an appropriate message. Ie., for a copy from a 29.5 Winchester disk to magtape the messsage is -

```
?SAVRES-I-57684 Blocks transferred to magtape
*
```

To exit from SAVRES and return to RT-11, type /Q (or CTRL/C) against the asterisk.

The system returns the monitor prompt.

The tape automatically rewinds to BOT and when ON-LINE light is extinguished, ie., the tape unit is off-line, and the UNLOAD light is on and not pulsating, the tape may be removed.

Using SAVRES for Winchester/Floppy disk Operation

These instructions are for a Save operation and assume running under RT-11. Where relevant, details of the Restore prompts are documented.

Running SAVRES

Load floppy disk Volume 1 in SF0:

Enter R SAVRES

The system displays -

```
SAVRES - V03.00
*
```

Enter the CSI command string:

For a Save operation from Winchester disk to floppy disk(s), the command is -

```
SF0:*=DM0:
```

For a restore operation from Floppy disk(s) to Winchester disk, the command is -

```
DM0:*=SF0:
```

For a Save operation the system displays -

Save DM0: onto SF0: are you sure?

For a restore operation the system displays -

Restore DM0: from SF0: are you sure?

If you type N (No), the system displays -

?SAVRES-F-Operation unsuccessful
. (monitor prompt)

If you type Y (Yes), the system displays -

Mount Volume 1 and hit RETURN key to continue.

If the incorrect volume is installed, the system displays:

You have mounted the wrong volume.
Please mount volume 1 in SF0: Hit RETURN to continue.

If no disk is installed, the system displays -

?SAVRES-F-Floppy disk size error
?SAVRES-F-Operation unsuccessful

SAVRES continues to copy to the diskettes and prompts the user to mount the next consecutive numbered volume.

If the RETURN key is pressed and no disk is loaded the system displays -

?SAVRES-W-Error reading home block

Insert the correct disk and enter RETURN.

On completion of a Save operation the system displays a message. For a 29.5 Mb Winchester disk to floppy disks operation the message is -

?SAVRES-I-57684 Blocks transferred to floppy
*

Type in /Q (or CTRL/C) against the asterisk prompt.

The system returns the monitor prompt.

Remove the last floppy disk from the drive.

INTRODUCTION

WUTIL, the Winchester Disk Utility, provides a number of functions for disk verification and maintenance. It must only be used with a patched DM (DEC RK07) disk driver (refer Section 7) as these patches modify the size of RK06/RK07.

```
+-----------------------------------------------------------------+
| IMPORTANT NOTE:                                                 |
|                                                                 |
|DEC's diagnostics MUST NOT BE USED on SPECTRUM Winchester disks. |
+-----------------------------------------------------------------+
```

WUTIL STARTUP MESSAGES

When WUTIL is run, the following messages are produced:

WUTIL V02.03c — Program i.d. and version number

WARNING — A warning message -

This utility is for Engineering Maintenance only. Misuse of it can easily corrupt disks.

```
Read  Buffer Address:  31000
Write Buffer Address:  71000
```

These messages are those of the program buffers. They may be examined or altered by the use of the 'BREAK' key to enter Console ODT. When alteration or examination is complete WUTIL can be continued by entering 'P' (for proceed).

```
  8.5Mb  8 inch H   or
   30Mb  8 inch G   or
   56Mb 14 inch K   or
  132Mb 14 inch J
```

This message is produced to inform the user as to what type of drive the controller senses is on the system. If it is not correct, check the controller switch settings. This message MUST BE CORRECT for the type of drive on the system.

A further message may occur during pattern testing -

This error indicated a serious drive problem
Please report it to Webster Electronics

This means that WUTIL has detected an error that should NEVER occur. The Design Department must be informed.

OPTION MENU

WUTIL produces the standard 'OPTION' menu and offers nine options:

1. Enter Interactive Mode.

   This allows a user to enter commands (generally of a single letter) to WUTIL, to be executed by the disk. This mode is described in detail in the paragraph 'Interactive Commands'.

2. Ignore I/O Errors

   Normally WUTIL regards I/O errors (except while pattern testing) as fatal and reports them to the user for action. However, if a "scope-loop" is needed to determine the exact cause of an error, the selection of this option will cause I/O errors to be ignored and WUTIL will continue execution.

3. Allow Writing and Formatting

   WUTIL does not allow it's interactive mode tests to destroy data unless this option has been selected.

4. Format a Disk (** CAUTION - DESTROYS ALL DATA ON THE DISK **)

   Before a Winchester disk can be used it must be formatted. This function formats the entire disk, not just one unit. The actual formatting is done while in unmapped mode (T-Board Switch 3 in ON position). While the disk is being formatted WUTIL displays the mapped cylinder counter which increments by 100 octal for each unit and the unmapped counter which increments by 22 octal for the whole disk.

5. Pattern Test (** CAUTION - DESTROYS ALL DATA ON THE DISK **)

As Winchester disks are non-removable, it is possible although unlikely that bad spots may be present on the disk. This function is designed to detect these bad spots on a 'per unit' basis (model K is configured as 2 x RK07 disks and model J as 4 x RK07 disks), and to accumulate statistics on a disk's errors. The statistics gathered by each test are then used by the 'Replace Bad Blocks' function to map out the bad parts of the disk. WUTIL specifies when a different unit has been selected and clears the Statistics Table. Unlike all other WUTIL functions, pattern testing continues indefinitely until interrupted by the user. The user must enter CTRL/C (twice in succession). WUTIL will then display requests; whether error statistics are to be displayed and whether the pattern test is to continue.

6. Display Pattern Statistics.

This function displays the Pattern Test error statistics gathered by the 'Pattern Test' option. These statistics are displayed on an 'ascending block-in-error' basis, with errors for that block being displayed under the appropriate heading. The following error types: DCK (15), OPI (13), DTE (12), ECH (06), SKI (01), are fully described in the Winchester Disk Programming Guide (Spectrum Eleven Hardware Manual). There are two remaining error types: CMP which is a data compare error caused when the data read does not match the data written, and HVR (Header Vertical Redundancy) which is not used on Spectrum Eleven machines.

7. Replace Bad Blocks (** CAUTION - DESTROYS DATA ON THE DISK **)

This option utilizes the Pattern Statistics table to decide if a block is bad whereupon it replaces it with the Winchester's spare block for the track. Even if there are no bad blocks on a unit, this option MUST ALWAYS be run after pattern testing as it writes the RK07 bad block file to disk. This file is needed by the RT-11, RSTS, and RSX-11M operating systems to determine where the bad blocks are. All Winchester models excepting the 8.5 Mbyte model H have at least one spare block per track. With the H models, WUTIL 'MARKS AS BAD' those blocks with bad error statistics. An INIT DMn:/REPLACE must be performed to replace these bad blocks in the disk's bad block replacement table in Block 1 of the disk. A block is determined to be bad if it has more than 10 errors in any one category or more than 20 errors in total for the one block.

10. Make Winchester DEC-Sized RK07.

    This option provides the capability for the Winchester models G, J, and K to exactly emulate an RK07 which is necessary for the RSX-11M device driver when using this DEC Operating System. This option is selected before the disk is formatted and bad blocks replaced, as it sets up the exact RK07 disk sizes in place of the SPECTRUM Winchester disk sizes.

11. Return to RT11.

    Exit from the WUTIL Utility to return to RT11.

DISK VALIDATION PROCEDURE

The following procedure should be used to ensure that a Winchester Disk is ready for customers use.

1. Format the disk using WUTIL.

2. Pattern test the disk, at least overnight. This is necessary to obtain good error statistics.

3. Whether the disk had any bad blocks or not, perform a Bad Block Replacement. This option is essential as it writes a bad block table onto the disk. RT-11 (and RSX-11M) require a valid table.

4. Prepare the Winchester for RT-11 use by:

   a. Initializing the disk
      INIT DM0:

   b. Copy across RT-11
      COPY /SYS SF0: DM0:

   c. Copy the bootstrap
      COPY/BOOT DM0:RT11SJ DM0:

   d. Boot the Winchester
      BOO DM0:

   e. If necessary, modify the STARTS.COM file to correct any assignment of DK:

INTERACTIVE COMMANDS

WUTIL provides an interactive Winchester diagnostic capability. It enables the user to enter commands to the disk. Multiple commands (as many as will fit on one line) may be entered. These commands are executed when a carriage-return is entered. The commands last entered are repeated if only a carriage-return is entered.

An example of a command is: *10CS

where -

*	the WUTIL command prompt
10C	set up 10 (octal) in the cylinder register
S	seek to the desired cylinder

A more complex command: *10CS400CS

where -

*	Wutil prompt
10C	Set up 10 as the desired cylinder
S	Seek to it
400C	Set up 400 as the desired cylinder
S	and Seek to it

This sequence may be repeated indefinitely or a specified number of times simply by enclosing it in brackets.

Example: *20<10CS400CS>

Repeat the sequence enclosed in brackets 20 (octal) times. If no number is specified, the sequence will be repeated indefinitely until CTRL/C is entered.

Example: *<10CS400CS>

This sequence will repeat indefinitely.

WUTIL COMMANDS

Where (Num) precedes a command, a number must be supplied. All numbers are entered in octal and if not, will display the message 'Illegal Symbol'. If no number is supplied the default used is 0 (zero).

ESC	Enter WUTIL's Option mode.
(Num)A	Set up the word count register (RKWC) with the two's complement of the specified number.
(Num)B	Set up the sector (or block) register (RKDA) with the specified sector address.
(Num)C	Setup the desired cylinder register (RKDC) with the specified cylinder.
D	Initiate a Power Down command (RKCS1<07>) in the specified drive.
E	Erase (zero) the Read buffer.
F	Format the specified cylinder.
G	Compare data in the Read and Write buffers.
(Num)H	Sets up the specified head (RKDA). A messsage is displayed if an illegal head number is entered.
I	Initialise Controller and Drives.
J	Reset any I/O errors that resulted from the last operation. I/O operations cannot be done unless the error bit has been reset. This allows an interactive command string to set up a scope loop, to continually set errors but to keep retrying the command. The 'Ignore I/O Errors' option must be selected prior to this command otherwise WUTIL will exit the loop reporting the error.
M	Move data from the Read to the Write buffer by doing a direct memory to memory copy.
(Num)N	Select the specified drive (RKCS2). On 8" Winchesters, only 0(zero)N may be entered. Although 1, 2, and 3 are accepted, these are reserved for 14" Winchesters. If a number other than 0 is used for 8" disks, command errors will display for commands R, S, V, and W, fatal error will display for Option 5, and I/O error will display for option 7. If an incorrect drive number has been specified, initialize the disk using command I and start again.
Q	Perform a Write Check function.
R	Execute READ command (21) from the specified address (cylinder, head, and sector). Number of words to be read given in RKWC. If 0, one block (256 words) assumed.

S	Execute SEEK command (17) to the specified cylinder.
U	Execute Power up (11) of the specified drive.
V	Execute Read format (35) of specified cylinder.
W	Execute Write data (25) to the specified disk address (as for Read above).
Xrr	Examine the specified register. A two letter register mnemonic must be supplied. Refer to paragraph 'Register Names' for details.
Z	Restore (recalibrate) specified drive (13 - RKCS1).
(Num)<...>	Execute the sequence inside the brackets the specified number of times. Where no number or zero is specified, the sequence is repeated indefinitely.

ADVANCED WUTIL COMMANDS

Wutil has five commands that enable numeric registers to be set, incremented, decremented, used as numeric values, or displayed. There are 26 numeric registers; these being referred to by the UPPER CASE letters A - Z. The register commands are:

Command	Description
#	The following letter is a numeric register of which the contents are to be used as a numeric value. Eg., #AC Place the contents of Register A in the Cylinder Register.
+	Add the current numeric value to the specified register. Eg., 1+AC Add 1 to the contents of Register A and move that value to the Cylinder Register.
-	Subtract as above. Eg., 1-AC Subtract 1 from Register A and move that value to the Cylinder Register

= Set the numeric register with the current number. Eg.,

1=A

Set the register A to have the value 1.

! Display the contents of the specified numeric register. Eg.,

!A

Display the contents of Register A.

Additionally, there is a further command that allows the addressing of Device registers.

@ Treat the following two letters as a Device Register name. Eg.,

10=@C1

Set the Control and Status Register 1 to 10 octal.

Complete Example

*0=A1400=B700<#ACS#BCS1+A1-B!A>

Set up a butterfly seek over the whole disk.

REGISTER NAMES

The following registers may be examined (displayed) via the X command:

Mnemonic	RK07 Name	Description
BA	RKBA	Data address for Read and Write.
C1	RKCS1	Control and Status Register 1. All status bits are displayed in mnemonic form.
C2	RKCS2	Control and Status Register 2. Status bit display.
DA	RKDA	Disk Address (track and sector).
DC	RKDC	Desired Cylinder.
DS	RKDS	Drive Status. Status bit display .
ER	RKER	Error Status. Status bit display.
WC	RKWC	Word Count for Reads and Writes.

INTRODUCTION

RANDY, a universal disk exerciser, provides an effective means for checking random access device hardware, ie., floppy disks, floppy disk drives, winchester disks, controllers, the iterator, etc. It offers a self-help diagnostic check for establishing faults and/or provides proving checks for new disks, drives, etc. Extreme care must be taken when using RANDY as existing data is destroyed. Files to be secured must either be copied to another device or be written to the outer most half of the disk. If the latter, the parameter option (detailed below) cannot be accepted as RANDY would operate over the whole disk.

Where a Floppy Disk drive or diskette is suspect, it is recommended initially that the floppy disk utility FUTIL be run with the 'Format & Verify' and 'List Bad Blocks' options being selected. If RANDY then displays persistent errors, Webster Electronics Service Department should be contacted.

This utility displays on the directory of the system disk as RANDY.SAV. A maximum of 4 logical units (DK0 to DK3) may be tested at once, with the option provided to test the inner half only rather than the full disk, as errors or faults are generally less probable in the outer half of a disk. This allows programs and data to be retained on the disk while testing is in progress. RANDY recognises DK units, and any combination of random access devices may be assigned to the logical units DK0 through DK3. If no assignments are entered, the system device is used as the default. Before starting the program, all other devices desired to be tested MUST be assigned to unique logical units, using the ASSIGN command.

PREPARATIONS PRIOR TO USING RANDY

1. Ensure that the information on the device is backed up or that it exists only on the outer half of the disk. If the latter, DO NOT allow RANDY to test the whole disk as all data will be destroyed.

2. Establish the number of units to be tested in the one run (maximum of 4 allowed) and then assign each device a unique sequential logical unit number.

As an example, the disks on a GC machine (1 Winchester and 1 Floppy) are to be tested with RANDY.

DM0: is automatically assigned to DK0: in the start file (STARTS.COM) during bootstrap procedures. SF0: must then be assigned by the user to DK1: (being the next sequential unit number).

Ie, ASS SF0 DK1

Note

Do not assign more than one device to the same unit number, and do not assign a device to a unit number which is not the next sequential number.

USING RANDY

This example assumes testing for two previously formatted floppy disks using DK0 and DK1.

1. Ensure that the system disk is inserted and type R RANDY

   The system displays -

   RANDY disk exerciser version 5
   ATTENTION! Install scratch disks now.

   Normally tests inner half of DK0 & DK1
   Parameter change required?

2. Remove the system disk and insert the formatted disks before acknowledging the parameter request. A parameter change gives the option to test the full disk, and/or to select the unit(s) to be tested.

   If N to the parameter change request, the system displays:

   Unit 0 Volume blocks 2464 Test blocks 1232
   Unit 1 Volume blocks 2464 Test blocks 1232
   Start testing?

   If Y, RANDY will commence to test the inner half of DK0 and DK1.

   If N, the original parameter request is displayed.

   If a parameter change is required for either option, enter a Y.

   The system will then display -

   Unit 0?

   Enter either Y or N, whatever is required.

   The system displays -

Whole Disk?

Enter the required response.

This dialogue is repeated for each unit assigned.

RANDY TESTS

RANDY progresses through each unit in turn before commencing the next test and will repeat the series of tests until instructed by the user. The tests are as follows:

Pass 1

Each test is displayed with the start time of the run.

Sequential write test:

The first four words of each block are written with dynamic information, viz,

WORD 0 : Unit number
WORD 1 : Block number
WORD 2 : Pass number
WORD 3 : Test number of pass

The remaining 252 words per block are filled with a standard pattern of text which includes a set of % signs which tend to form a worst case binary pattern for data recovery circuits.

Testing disc integrity:

Sequentially reads the entire area of the disk being tested, exercising the hardware integrity checking logic. If no hardware errors are detected, the recovered data is compared with expected. If there are errors in any of these testing sequences, the system will display -

Unit (0) Block (2431) Disc integrity error ***

When a data compare error is detected, the expected and received data are displayed in octal, together with the word number of the block in error. If the word number has a value 0-3, various addressing problems may be inferred by reference to the meanings of these words discussed above.

Unit scan write test:

A block at a time is written to each unit, in a continuous rotation, until all blocks specified for each unit have been written to. This is followed by a disk integrity test.

Butterfly write test:

This test produces considerable exercising of the head positioning mechanism by writing first the outermost, then the innermost block, progressing through all the even-numbered blocks in ascending order alternating with the odd-numbered blocks in descending order, until all blocks have been written. This is followed by a disk integrity test.

Random write test:

A random number generator selects the blocks and disks at random and writes the test pattern text 5000 times.

On completion of these tests, statistics of the pass are displayed, ie.,

Pass Statistics

DK0	DK1	DK2	DK3	
16	16	0	0	Sequential writes
1232	1232	0	0	Unit scan writes
1232	1232	0	0	Butterfly writes
2489	2511	0	0	Random writes (totals 5000)

In this example, both DK0 and DK1 were tested on the inner half of the disk only. Although only two units were tested, the remaining available unit headings are displayed.

Cumulative error statisticsa

DK0	DK1	DK2	DK3	
0	0	0	0	Seq. write errors
1	0	0	0	Seq. integ.errors
0	0	0	0	Uscan write errors
0	0	0	0	Uscan integ. errors
0	0	0	0	B'fly write errors
2	0	0	0	B'fly integ. errors
0	0	0	0	Rand write errors
0	0	0	0	Rand integ. errors
0	0	0	0	Data Comp. errors

In this example, DK0 is displaying faults. Assuming that the disk has proved itself on FUTIL and has been formatted, it suggests that the drive is faulty.

COMPLETING RANDY

The above routine will process continuously with the Pass number incrementing by one each time a sequence is finished. When proving new disks, it is recommended that this utility be run overnight as each pass takes approximately 35 minutes for a two drive floppy disk system.

When testing is suspended by typing double CTRL/C, an error report is displayed, followed by:

Continue testing?

If Y, RANDY will continue from the point of suspension.

If N, the system displays -

'Return to Monitor?'

To exit completely from RANDY, it is necessary to remove the disk inserted for testing and re-install the System disk. Only then, type in Y. If N is typed, a return is made to the parameter selection and restart routines.

OVERVIEW

WARNING - these procedures are for Engineering Maintenance only. Notify Webster Electronics Customer Service division of any Winchester Disk failure.

Two of the Spectrum Eleven models (KM and JM) incorporate magnetic tape instead of floppy disks as the backup media. Software is provided to allow these machines to be supported with only a magnetic tape if a Winchester failure occurs, or when they are being built.

This chapter gives details on constructing a bootable magtape, with further instructions for creating the primary bootstrap file, booting the magtape, creating a bootable disk, and restoring a disk. Whether a saved disk is to be restored, or a basic RT-11 system is to be installed on a disk, the Winchester Disk utility (WUTIL) is first used to format and pattern test the disk. During formatting the whole disk is formatted but pattern testing, ALWAYS followed by the 'Replace Bad Blocks' function, is be done on each unit in turn. Refer WUTIL for full details.

CREATION OF A BOOTABLE MAGTAPE

An additional device handler (SM) has been provided to allow RT-11 to be booted and run from magnetic tape. Refer to the paragraph on the Spectrum Magtape handler (Section 7) for details. A bootable magtape is created as follows:

First, create a disk image of the required RT-11 system, including the SM.SYS handler.

Put a bootstrap on the disk image with the SM primary boot. Ie., assuming the image is on a virtual unit (VU6:) -

```
COPY/BOOT:SM VU6:RT11SJ VU6:
```

Now create the tape:

Initialize the magtape with the special bootstrap file (refer next paragraph) -

```
INIT MT0:/FILE:SMBOOT.BOT
```

Copy the disk image to tape as a file. Note that the image MUST be the FIRST file on the tape. Other files may be on the tape if required. -

```
COPY/DEV/FILE VU6: MT0:SYSTEM.FIL
```

A bootable magtape now exists. The structure is:

VOL1 SMBOOT HDR1 * data * EOF1 * *

where:

VOL1	volume label
SMBOOT	bootstrap, loaded and executed by hardware bootstrap
HDR1	SYSTEM.FIL ANSI file label
*	tape mark
data	the disk image, in 512 byte blocks
*	tape mark
EOF1	the End-of-File label

MAGNETIC TAPE BOOT FILE

In the INIT command for creating a bootable magtape (previous paragraph), is a switch '/FILE:SMBOOT.BOT'. This switch causes the specified file to be written to the magtape as the second block. The hardware bootstrap always loads into location 0 (zero) and executes the contents of the second block on the magtape. This block must therefore contain the primary bootstrap which is file 'SMBOOT.BOT'.

Because the primary bootstrap is only a portion of the bootstrap written to the start of the disk image by the 'COPY/BOOT' command (in fact just the first block of the disk image file), SMBOOT is created from that disk image:

First take a copy of the disk image:

```
COPY/DEV/FILE VU6:DM0:SMBOOT.BOT
```

This takes a copy of the disk image. DM0: is a practical place as this file is reasonably large even though it is only a temporary file.

Now it is made into a one block file -

Find out where it starts:

```
DIR DM0:SMBOOT.BOT/BLOCK
```

then . . . .

```
DELETE DM0:SMBOOT.BOT
```

followed by:

```
CREATE DM0:SMBOOT.BOT/START:xxxxx/ALLOCATE:1
```

This truncates the file down to one block which contains the primary bootstrap written to the disk image by the 'COPY/BOOT' command.

This file may then be used when initializing the magtape (INIT MT0:/FILE:SMBOOT.BOT)

BOOTING THE MAGTAPE

1. Load the magtape. Press 'LOAD', 'ONLINE'.
2. The 'LOAD' and 'ONLINE' lights will flash. When they stop flashing the tape is at 'Load Point'.
3. Press in the Winchester 'ACCESS' button to disable the Winchester disk.
4. Press the 'RESTART' button. Note that the magnetic tape can ONLY be booted when it is at 'Load Point', otherwise a system crash will occur. The 'LOAD' light will go off and RT-11 will be booted from magnetic tape.

CREATING A BOOTABLE DISK

When RT-11 has been loaded, you may then create a bootable disk. The tape has all the files necessary to run a small RT-11 system.

1.	INIT DM0:	to initialize the disk
2.	COPY/SYS SM0: DM0:	copy all files to the disk
3.	COPY/BOOT DM0:RT11SJ DM0:	put the bootstrap on the disk
4.	BOO DM0:	and boot off the disk.

RESTORING A DISK

If it is not possible to boot from the disk it may be just the disk bootstrap that is destroyed or all information on the disk. In either case, using the RT-11 magnetic tape can get back a working disk. How recent the information is depends on the frequency that the Save/Restore utility (SAVRES) has been run.

First, bring up RT-11 on magtape (as described above) and then attempt a directory of the destroyed disk. If this works it signifys that it is only the bootstrap on the first block of the disk that has been destroyed. This can be fixed by using the COPY/BOOT command. If it is not possible to get a directory, proceed as follows:

1. R SAVRES — Run the Save/Restore utility

2. * — DM0:*=MT4:

   Replace DM0: (or whatever winchester unit is to be restored) by MT4: (streaming mode). MT0: is the slow magtape mode)

3. Restore DM0: from MT4: are you sure?

   Y (yes)

4. SAVRES tape must now be loaded. Hit RETURN to continue.

   Press 'UNLOAD' to unload the RT-11 tape. Load the SAVRES tape and press 'LOAD', 'ONLINE'. When the tape is at 'Load Point', enter RETURN.

The magtape will then be restored to disk which takes approximately 3 minutes. When the end message is displayed, the system will crash because the system device has been used for the SAVRES tape which is not allowed by RT-11. This cannot be avoided but since a bootable disk has now been built, it should be possible to boot from that disk. However, if a bootable disk has not been built, press 'ONLINE' (which takes the tape offline), 'UNLOAD' to unload the SAVRES tape, and remove the tape. Load the RT-11 tape and 'LOAD', 'ONLINE' to bring the tape back to 'Load Point', then re-boot the magtape. An RT-11 system is now available to work with.

INTRODUCTION

Most of the devices on SPECTRUM systems are supported by standard DEC RT-11 device handlers. However, handlers specifically written or adapted for SPECTRUM peripherals which do not conform to DEC standard devices, are described in this chapter.

1.1 CARD READER HANDLER

1.2 ITERATOR HANDLER

The iterator handler, IT.SYS, enables the iterator to simulate an extremely rapid disk by defining areas of extended memory as a file-structured device. This is particularly useful as a user swapping device for TSX, as a system device for rapid access to monitor swaps and overlays, or as storage for a highly overlaid program. By comparison with a disk, the iterator has effectively zero access time and a continuous transfer rate of 1.3 megabyte per second.

The validity of this particular use of the iterator is confined to SS11 models, as all of the above functions are handled more efficiently still in SS23 models through use of memory management and the TSX-Plus operating system.

The iterator handler automatically acquires all available memory above location 170000, and the effective volume size depends on memory size as follows:

Memory Size	IT Volume Size
64 kilobyte	8 blocks (4 kilobyte)
128 kilobyte	136 blocks (68 kilobyte)
256 kilobyte	392 blocks (196 kilobyte)

In order for the system software to automatically send the IT volume size, the iterator handler responds to the standard volume size system call .SPFUN 373.

As the iterator device is volatile, its directory and contents must be restored at each power-up. This may of course be achieved automatically through an indirect start command file. To initialize the IT handler, enter the command:

.INITIALIZE/SEGMENTS:1 IT:

If it is now desired to improve performance of an overlaid program such as BASIC or DUNGEON, simply COPY it to IT:, then RUN IT:program.

If it is desired to use the iterator as the system device in order to improve monitor responsiveness in single user applications, simply COPY/SYS all desired system files to IT: , install a bootstrap using the COPY/BOOT command, then BOOT IT:

If it is desired to use the iterator as a rapid user swapping device for TSX (limited to 3 or 4 users), it is only necessary to edit the TSX file TSGEN.MAC to address IT: as the swap device, and include IT: in the AUTHAN specification, and then rebuild TSX.

1.3 FLOPPY DISK HANDLER

The Webster Electronics Floppy Disk controller is in two parts; the GPMI interface which handles the data flow between the floppy disk and memory, and the Western Digital FD1781 Floppy Disk controller chip which manages the disk.

The Floppy Disk handler (SF) is able to bootstrap and run in all SPECTRUM models and handles all formats which the model is capable of accepting. File oriented operations can be performed upon alien formats without the installation of a special handler and transfers between any two alien formats can be effected by removing the System disk and using the RT-11 COPY/WAIT command.

Variable Size Volume Capability

The size of the volume installed is returned to the RT-11 Utilities or to any MACRO or FORTRAN program, via the system call .SPFUN 373 and it's function exactly emulates that of the DEC DY handler.

Read-after-Write

A set option SET SF [NO]RAW is built into the handler to allow optional read-after-write checking. If set to the RAW state, upon completion of every WRITE operation, all written blocks are re-read to verify cyclic redundancy integrity. Upon CRC failure, the WRITE is repeated from the failure point to the

end of the transfer and is again re-read. Failure of the second re-reading is reported as a hard error. After each setting or un-setting of RAW, the system must be re-booted before the option will take affect. The default state of the handler on distribution will be NORAW.

SF Handler Speed Enhancements

Previous versions of the SPECTRUM Floppy Disk handlers (DD, DF, DW, DX) had substantial overhead at the entry point which was redundant under many conditions. On entry, the SF handler checks if the unit is unchanged and whether the head is still engaged from the previous operation. This assumes that FDTR and FORMAT are still valid, allowing the HEAD UNLOAD and READ ADDRESS format sensing operations to be skipped. As well, if the track address is unchanged, the SEEK operation is avoided.

Consequently, there are no unnecessary operations performed at handler entry. For example, when sequential reads or writes are called from the application as a series of single-block transfers, it should theoretically be possible to do this without losing a revolution for each block. However, monitor overheads are such that this may only be achieved by use of interleaved sector addressing formats.

Control of RETRY Action

Upon discovery of a potentially recoverable (soft) error, an action word (normally 052525), is right shifted to determine recovery procedure. When a 1 is shifted out of the rightmost bit, the heads are stepped outwards one track before the operation is retried. A zero results in step in. When all the bits are finally shifted out of the action word, a homeseek and retry is attempted, failing which a hard error is reported to the monitor.

This action is altered by the keyboard command SET SF ACTION=n, where n is the octal action bit pattern desired.

Example patterns are:

n	Error Recovery Action
0	No retry action, all soft errors reported
1	Homeseek and one retry
2	Stepin-retry, then homeseek-retry
5	Stepout-retry, stepin-retry, homeseek-retry
12	Stepin, stepout, stepin, homeseek
52525	Alternate stepout/in for 14 retries, homeseek
177777	Stepout and retry endlessly until recovery

After setting SF ACTION=n, the system must be re-booted before the option will take effect.

The following SPFUN requests are supported and their functions are as for the DEC DY handler.

```
000373 SIZ$FN =: 373  ;Return volume size
000375 WDD$FN =: 375  ;Write physical sector, del. data mark
000376 WRT$FN =: 376  ;Write physical sector
000377 RED$FN =: 377  ;Read physical sector
```

Null Filling of Partial Sectors

In the 'SS' Series (GPMI-S Controller) SPECTRUMs, null filling of partial sectors is automatically managed by the hardware. When extended memory is enabled (MMG$T=1), the presence of a GPMI-S is assumed and all software action for null filling is bypassed.

1.4 WINCHESTER DISK HANDLER

Although the Winchester disks emulate DEC's RK disks, they are different in size.

Webster Electronics provide customized DM handlers for RT-11 (DM.SYS) and TSX-Plus (DM.TSX). It is necessary to use these since there may be one of three different sizes of RK07 on a Spectrum.

These modified handlers determine the type of Winchester attached to a Spectrum and set up the correct size for the RT-11/TSX-Plus calls.

The sizes set up are:

Model	Size	Units	Disk Type
HC	16632 blocks	1 unit	RK06
GC	57684 blocks	1 unit	RK07
KM	55374 blocks	2 units	RK07
JM	64614 blocks	4 units	RK07

1.5 SPECTRUM MAGTAPE HANDLER

This is a Webster Electronics Block structured MT handler. This handler treats a magtape as a read-only block-structured device. It allows RT-11 to be run from magtape. HOWEVER, it must be appreciated that the SM handler was only developed to provide a magtape bootstrap and diagnostic capability. It is NOT intended to provide full RT-11 capabilities, and loses position on the magtape if CTRL/C CTRL/C is used at inconvenient times.

This handler is a modification of DEC's FSM.MAC. It allows a magtape to be used as a file structured device. This handler MUST BE CALLED SM: If it is not, a crash with I/O errors will occur. This limitation is imposed by the primary boot routine. To compile this handler:

```
R MACRO
MTBLOK=SYCND,SYMAGT,MTBLOK
CNTRL-C
```

To link:

```
LINK /EXEC:SM.SYS TM,MTBLOK
```

Note that TM is the standard DEC MT hardware handler. It must have been compiled with MT$FSM = 1 (to include the MTBLOK interface).

SM will then allow random access reads to a magtape. The magtape must be of the form:

VOL 1 etc * data (disk image, 512 byte blocks) * EOF1 * *

where * indicates a tape mark.

SM, when it first starts, always rewinds the tape to the beginning then positions the tape after the first tape mark. To create an SM readable magtape:

INIT MT0:	initialize it
COPY/DEV/FILE	create a disk image on the tape
From? VU6:	a virtual unit is a fine place for a disk image
To:MT0:SYSTEM.FIL	any file name will do, but a filename MUST BE SUPPLIED.